ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Εργαστήριο Μικροεπεξεργαστών και Υλικού

"ΜΕΛΕΤΗ, ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΣΕ ΑΝΑΔΙΑΤΑΣΣΟΜΕΝΗ ΛΟΓΙΚΗ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ CE ΓΙΑ ΣΥΓΚΡΙΣΗ ΠΡΩΤΕΪΝΩΝ ΩΣ ΠΡΟΣ ΤΗΝ ΤΡΙΤΟΤΑΓΗ ΤΟΥΣ ΔΟΜΗ"

Καγιαβάς Χρίστος

Επιτροπή:

Δόλλας Απόστολος, Καθηγητής (Επιβλέπων)

Πνευματικάτος Διονύσιος, Αναπληρωτής Καθηγητής

Παπαευσταθίου Ιωάννης, Επίκουρος Καθηγητής

Χανιά 2009

Περίληψη

Οι πρωτεΐνες είναι μακρομόρια που εκτός από τον ρόλο του δομικού στοιχείου του κυττάρου που έχουν, έχουν και τον σημαντικό ρόλο της ρύθμισης και της εκτέλεσης των βιολογικών διεργασιών. Ο κύριος παράγοντας που επηρεάζει αυτή την ιδιότητά του είναι η μοναδική τους στερεοδομή που καθορίζεται από την τριτοταγή τους δομή. Το γεγονός αυτό οδήγησε στην ανάπτυξη διάφορων μεθόδων σύγκρισης πρωτεϊνών ως προς την τριτοταγή τους δομή με στόχο τον εντοπισμό ομοιοτήτων στην λειτουργία τους.

Ένας αλγόριθμος που αναπτύσσει με ταχύτητα και ακρίβεια την σύγκριση αυτή μεταξύ δύο πρωτεϊνών είναι ο αλγόριθμος CE που περιλαμβάνεται στο Bioperf, ένα πακέτο-σημείο αναφοράς, με υλοποιήσεις αλγόριθμων βιοπληροφορικής. Για τους λόγους αυτούς επιλέχθηκε ο συγκεκριμένος αλγόριθμός, μελετήθηκε και στην συνέχεια σχεδιάστηκε και υλοποιήθηκε σε αναδιατασσόμενη λογική ένα μεγάλο και βασικό μέρος του αλγορίθμου επιδιώκοντας κάποιο κέρδος όσον αφορά τον χρόνο εκτέλεσης του αλγορίθμου.

Ευχαριστίες

Η εργασία αυτή εκπονήθηκε στο Εργαστήριο Μικροεπεξεργαστών και Υλικού του Πολυτεχνείου Κρήτης υπό την επίβλεψη του Καθηγητή κ. Απόστολου Δόλλα, κατά την διάρκεια του ακαδημαϊκού έτους 2008-2009.

Θα ήθελα πρώτιστων να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Απόστολο Δόλλα για την υποστήριξή του, την καθοδήγηση και τις συμβουλές του αλλά και την εμπιστοσύνη που έδειξε στις δυνατότητές μου. Επίσης θα ήθελα να ευχαριστήσω τον Αναπληρωτή Καθηγητή κ. Διονύσιο Πνευματικάτο και τον Επίκουρο Καθηγητή κ. Ιωάννη Παπαευσταθίου για την συμμετοχή τους ως μέλη της εξεταστικής επιτροπής.

Ένα μεγάλο ευχαριστώ επίσης για τους Διδακτορικούς Φοιτητές Ευριπίδη Σωτηριάδη και Γρηγόριο Χρυσό, για την πολύ μεγάλη βοήθειά τους και την καθοδήγησή τους σε κάθε βήμα της πορείας από την αρχή μέχρι και την ολοκλήρωση αυτής της εργασίας.

Τις ευχαριστίες μου επίσης θα ήθελα να εκφράσω στον υπεύθυνο του εργαστηρίου Μικροεπεξεργαστών και Υλικού κ. Μάρκο Κιμιωνή, καθώς όλα τα μέλη του εργαστηρίου για την βοήθεια και την φιλία τους.

Για την φιλία και την στήριξή τους θα ήθελα να ευχαριστήσω και όλους τους φίλους και συναδέλφους που ήταν δίπλα μου τόσο όχι μόνο στις σπουδές αλλά και στη ζωή.

Τέλος θα ήθελα να πω ένα μεγάλο ευχαριστώ και να εκφράσω την αγάπη και την ευγνωμοσύνη μου στους γονείς μου που πάντα ήταν δίπλα μου και με στηρίζουν στα όνειρα και τις αποφάσεις μου.

στους γονείς μου, Στέλιο και Ελένη
και στην αδερφή μου Αλεξία

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή **1**

1.1 Εισαγωγή στην σημασία της τριτοταγούς δομής για τις πρωτεΐνες και τον αλγόριθμο CE 1

1.2 Δομή της διατριβής 3

ΚΕΦΑΛΑΙΟ 2

Πρωτεΐνες, η Τριτοταγής δομή και διάφορες μέθοδοι σύγκρισής της **5**

2.1 Πρωτεΐνες και η τριτοταγής δομή 5

2.1.1 Οι δομικοί λίθοι των πρωτεϊνών 5

2.1.2 Τριτοταγής δομή πρωτεϊνών 7

2.1.3 Τα περιεχόμενα ενός PDB αρχείου 8

2.2 Διάφορες μέθοδοι σύγκρισης της δομής των πρωτεϊνών 10

ΚΕΦΑΛΑΙΟ 3

Ο Αλγόριθμος CE **14**

3.1 Μέθοδος Combinatorial Extension (CE) 14

3.2 Ανάλυση αλγορίθμου Combinatorial Extension (CE) 15

3.3 Μελέτη και μοντελοποίηση του αλγορίθμου CE με την χρήση της γλώσσας προγραμματισμού C 21

ΚΕΦΑΛΑΙΟ 4

Σχεδίαση και Υλοποίηση της Αρχιτεκτονικής για τον Αλγόριθμο CE **24**

4.1 Εισαγωγή στην σχεδίαση και την υλοποίηση 24

4.2 Βασικές δομικές μονάδες της σχεδίασης και της υλοποίησης 25

4.2.1 Επιλογή της FPGA και του λογισμικού 25

4.2.2 Αναπαράσταση αριθμών κινητής υποδιαστολής 26

4.2.3 Μονάδες εκτέλεσης πράξεων κινητής υποδιαστολής απλής Ακρίβειας 26

4.3 Μνήμη BRAM 33

4.3.1 Μνήμης MEM_Protein1 και MEM_Protein2 33
4.3.2 Μνήμη Dnn 34
4.3.3 Μνήμη next61 35
4.3.4 Μνήμη Path 37
4.4 Μονάδα υπολογισμού ευκλείδειας απόστασης 38
4.5 Μονάδα υπολογισμού του D_{nn} και σύγκρισής του με το D_o 40
4.6 Μονάδα υπολογισμού και ελέγχου του κριτηρίου επιλογής του καλύτερου AFP από τα υποψήφια για επέκταση του μονοπατιού ομοιότητας 47
4.7 Βελτίωση της σχεδίασης για εξοικονόμηση των πόρων που έχουν κοινή χρήση από διαφορετικές μονάδες 51
4.8 Περιγραφή της μονάδας Fill Dnn memory, για την εγγραφή της μνήμης Dnn 54
4.9 Περιγραφή της μονάδας Select the Bit, για την επιλογή του bit της μνήμης Dnn που αντιστοιχεί σε συγκεκριμένο AFP 54
4.10 Μονάδα ελέγχου 54
4.10.1 Πρώτη φάση λειτουργίας 55
4.10.2 Δεύτερη φάση λειτουργίας 57

ΚΕΦΑΛΑΙΟ 5
Έλεγχος λειτουργίας και Αποτίμηση της απόδοσης 65
5.1 Έλεγχος λειτουργίας 65
5.2 Απόδοση συστήματος 67
5.3 Αποτίμηση απόδοσης 67
5.4 Αξιοποίηση των αχρησιμοποίητων πόρων της FPGA με στόχο την βελτίωση της απόδοσης 74

ΚΕΦΑΛΑΙΟ 6
Συμπεράσματα και Μελλοντικές Επεκτάσεις και Αλλαγές 79
6.1 Συμπεράσματα 79
6.2 Μελλοντικές επεκτάσεις και αλλαγές 79

ΒΙΒΛΙΟΓΡΑΦΙΑ 81

1

Εισαγωγή

Στο πρώτο κεφάλαιο της παρούσας διπλωματικής εργασίας γίνεται μία εισαγωγή στον αλγόριθμό CE και στην σημασία της τριτοταγούς δομής των πρωτεϊνών που προσδίδει μια μεγάλη χρησιμότητα σε αυτόν. Επίσης γίνεται μία σύντομη περιγραφή σχετικά με την προσέγγιση που έγινε για την λύση του προβλήματος της διπλωματικής εργασίας και δίνεται μια περιγραφή του τρόπου της δομής που θα ακολουθηθεί.

1.1 Εισαγωγή στην σημασία της τριτοταγούς δομής για τις πρωτεΐνες και τον αλγόριθμο CE

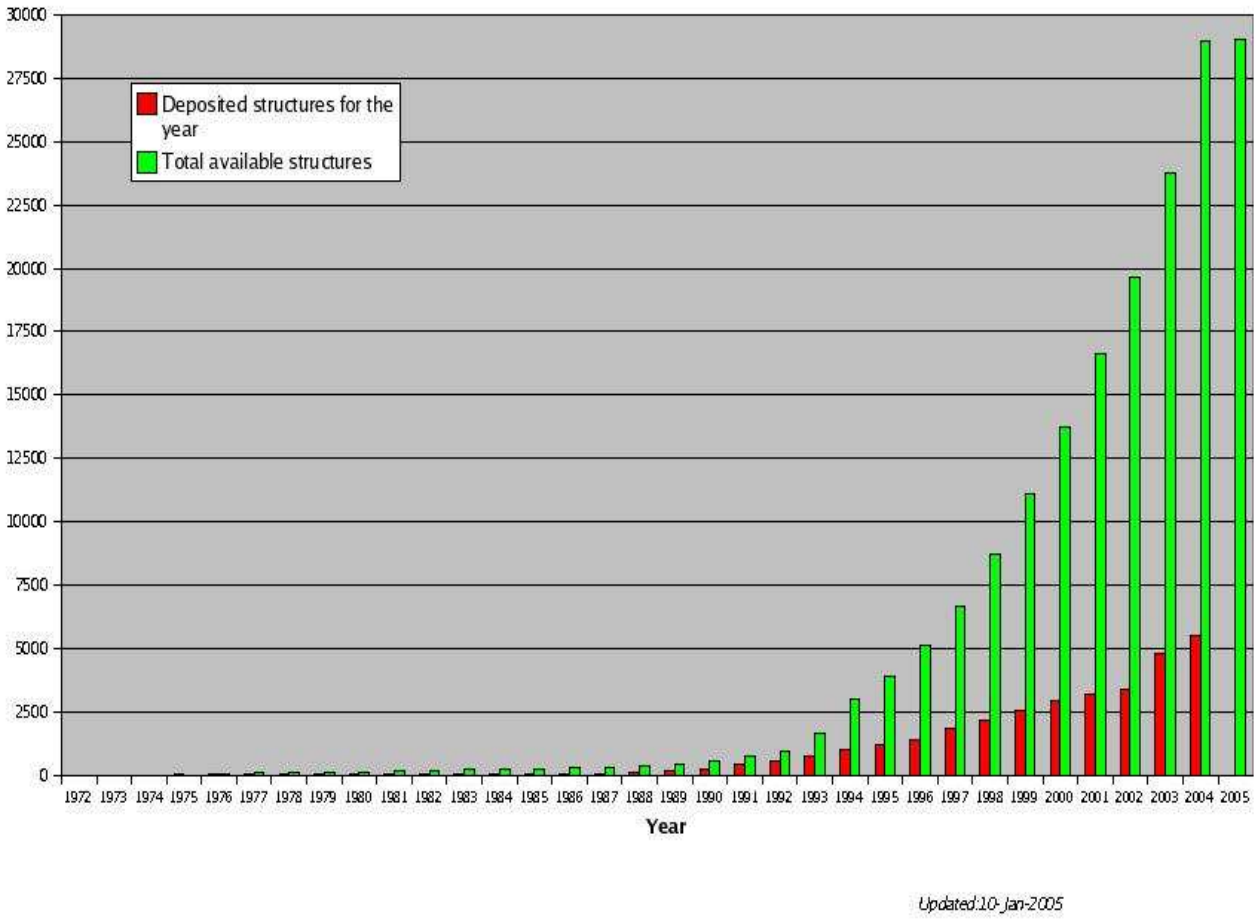
Οι πρωτεΐνες οφείλουν την σημασία τους στην μοναδική στερεοδομή τους. Αυτή τους επιτρέπει να δεσμεύουν ή να δεσμεύονται, να ρυθμίζονται και να ρυθμίζουν βιολογικές διεργασίες και να μεταφέρουν μικρότερα μόρια, να αλληλεπιδρούν και να αναγνωρίζουν επιλεκτικά μικρά ή μεγαλύτερα μόρια και να καταλύουν βιοχημικές αλληλεπιδράσεις. Η δομή των πρωτεϊνών σε αντίθεση με την αλληλουχία τους έχει την ιδιότητα να τείνει να διατηρείται διαμέσου της εξέλιξης και δύο ομόλογες πρωτεΐνες έχουν γενικά την ίδια δομή. Αντίθετα με την αβεβαιότητα που εμπνέει μια ομοιότητα στην αλληλουχία, η σύγκριση δύο πρωτεϊνών ως προς την δομή καθορίζει ορθά τον συσχετισμό δύο πρωτεϊνών και ως εκ τούτου αποτελεί μία πιο αξιόπιστη μέθοδο για τον εντοπισμό λειτουργικών ομοιοτήτων.

Ο αλγόριθμος ***Combinatorial Extension (CE)*** είναι ένας γρήγορος και ακριβής αλγόριθμος, με στόχο την εύρεση του βέλτιστου μονοπατιού ομοιότητας και είναι κατάλληλος για την εξέταση βάσεων και λεπτομερή ανάλυση μεγάλων πρωτεϊνικών οικογενειών. Για χρήση της εφαρμογής του *CE*, η οποία υποστηρίζεται στο San Diego Supercomputer Center (SDSC), καταφεύγουν περίπου 5000 ερωτήματα κάθε μήνα επί του παρόντος και έχουν κατεβαστεί πάνω από 300 αντίγραφα του λογισμικού. Παρ’ όλα αυτά σύμφωνα με μετρήσεις που έχουν γίνει για

αυτό τον αλγόριθμο αν θεωρήσουμε ως μέσο χρόνο υπολογισμού τα 30 δευτερόλεπτα ο συνολικός χρόνος για να γίνει σύγκριση μεταξύ όλων των πρωτεϊνών εκτιμάται σε μερικές χιλιάδες χρόνια.

Με κίνητρο την μεγάλη σχετικά χρονική διάρκεια εκτέλεσης του αλγορίθμου και την ολοένα πιο μεγάλη ζήτηση της εφαρμογής του, η παρούσα διπλωματική εργασία καλύπτει τον σχεδιασμό και την υλοποίηση μιας αρχιτεκτονικής σχεδιασμένης αποκλειστικά για την εκτέλεση του αλγορίθμου CE σε αναδιατασσόμενη λογική με κύριο στόχο την επίτευξη μιας σημαντικής επιτάχυνσης στον χρόνο εκτέλεσης του αλγορίθμου σε σχέση με ένα συμβατικό PC.

Η υλοποίηση της εργασίας ξεκίνησε με την μελέτη της σχετικής βιβλιογραφίας και του αλγορίθμου. Λόγω της πολυπλοκότητας του και της έλλειψης αναφορών και τεκμηρίωσης στις λεπτομέρειες υλοποίησης, έγινε αρχικά η υλοποίηση του αλγορίθμου σε λογισμικό (σύστημα αναφοράς) και στη συνέχεια σε αναδιατασσόμενη λογική συγκεκριμένων κομματιών του αλγορίθμου τα οποία αφορούν την κύρια διαδικασία εύρεσης ενός βέλτιστου μονοπατιού ομοιότητας μεταξύ των τριτοταγών δομών δύο πρωτεϊνών. Δεν έχει υλοποιηθεί καμία από τις διαδικασίες του αλγόριθμου που μπορούν να θεωρηθούν ως βελτιστοποιήσεις.



Σχήμα 1.1: Αύξηση Μεγέθους βάσης PDB (Protein Data Bank). Η ολοένα και μεγαλύτερη αύξηση των διαθέσιμων πρωτεϊνικών δομών κάνει εντονότερη την ανάγκη για σύγκρισή των δομών αυτών χρησιμοποιώντας μία γρήγορη και αποδοτική μέθοδο.
(*www.cs.ucy.ac.cy/courses/EPL450/lectures/**lecture1new.ppt***)

1.2 Δομή της διπλωματικής εργασίας

Στην συνέχεια η διπλωματική εργασία περιλαμβάνει ακόμα πέντε κεφάλαια. Στο Κεφάλαιο 2 μελετώνται οι πρωτεΐνες και η τριτοταγής δομή τους έτσι ώστε να ενημερωθεί ο αναγνώστης για κάποιους όρους σχετικούς με την δομή την πρωτεϊνών που είναι χρήσιμοι για την κατανόηση του αλγορίθμου. Επίσης γίνεται αναλυτικότερη αναφορά στην σημαντικότητα της τριτοταγούς δομής των πρωτεϊνών και αναφέρονται κάποιες μέθοδοι που χρησιμοποιούνται για την σύγκριση της τριτοταγούς δομής των πρωτεϊνών πέραν του αλγόριθμου CE.

Τα βήματα του αλγόριθμου CE παρουσιάζονται στο Κεφάλαιο 3 αναλυτικά και δίνεται η προσέγγιση του αλγορίθμου που έγινε κατά την μοντελοποίηση του και που θα χρησιμοποιηθεί ως βάση κατά την διαδικασία της σχεδίασης.

Στο Κεφάλαιο 4 καταπιάνεται με την σχεδίαση και την υλοποίηση της αρχιτεκτονικής για τον αλγόριθμο CE στην FPGA, ενώ το Κεφάλαιο 5 ασχολείται με τον έλεγχο τον αποτελεσμάτων για την πιστοποίηση της σωστής λειτουργίας καθώς και με την απόδοση του συστήματος που σχεδιάστηκε και υλοποιήθηκε.

Τέλος το Κεφάλαιο 6 περιέχει συμπεράσματα σχετικά με την απόδοση και ιδέες για μελλοντική επέκταση ή τροποποίηση της αρχιτεκτονικής.

2

Πρωτεΐνες, η Τριτοταγής δομή και διάφορες μέθοδοι σύγκρισής της

Το κεφάλαιο αυτό περιλαμβάνει το απαραίτητο βιολογικό υπόβαθρο. Εξηγούνται βασικές έννοιες, σχετικά με την δομή των πρωτεϊνών, οι οποίες είναι χρήσιμες για την κατανόηση του επόμενου κεφαλαίου. Γίνεται επίσης μία εμβάθυνση στο ζήτημα της σύγκρισης των πρωτεϊνών ως προς την τριτοταγή τους δομή και παρουσιάζονται διάφορες μέθοδοι σχετικές με αυτή.

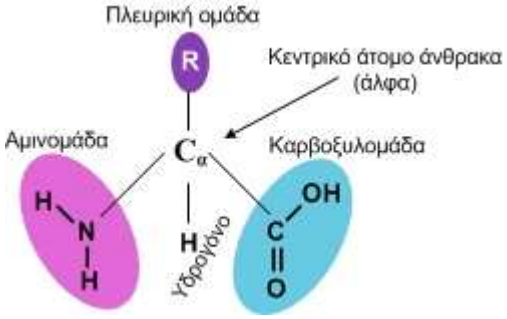
2.1 Πρωτεΐνες και η τριτοταγής δομή

Οι πρωτεΐνες αποτελούν τα ποιο διαδεδομένα και πολυδιάστατα, τόσο στη μορφή όσο και στη λειτουργία τους μακρομόρια. Οι πρωτεΐνες αποτελούν είτε το δομικό συστατικό του κυττάρου είτε επιτελούν πληθώρα δραστηριοτήτων σε αυτό. Ακόμα και στο πιο απλό κύτταρο εντοπίζονται εκατοντάδες διαφορετικές πρωτεΐνες και η κάθε μία από αυτές έχει ένα ιδιαίτερο ρόλο.

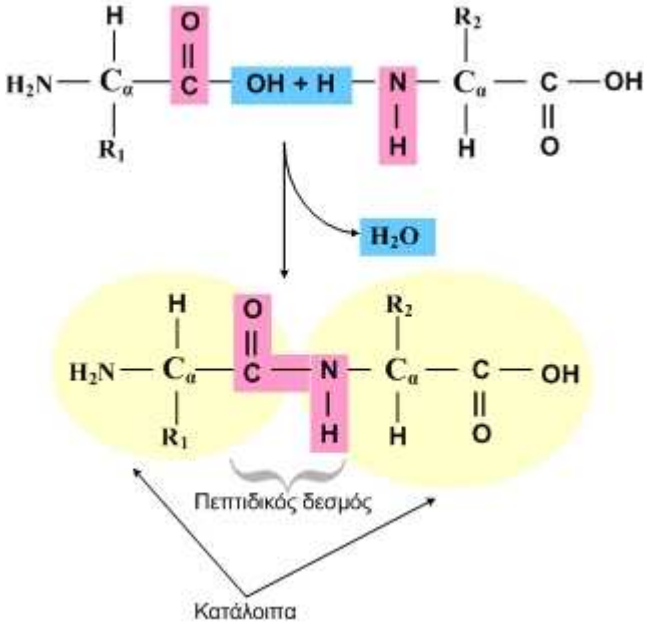
2.1.1 Οι δομικοί λίθοι των πρωτεϊνών

Οι πρωτεΐνες είναι πολύπλοκα μόρια αποτελούμενα από μία η περισσότερες αλυσίδες αμινοξέων. Κάθε αμινοξύ αποτελείται από ένα κεντρικό άτομο άνθρακα (***C_a***) που είναι συνδεδεμένο με μία αμινομάδα (NH_2), μια καρβοξυλομάδα (COOH), ένα άτομο υδρογόνου (H) και μία πλευρική ομάδα (R). Για την δημιουργία μιας πολυπεπτιδικής αλυσίδας από αμινοξέα η καρβοξυλομάδα ενός αμινοξέος δημιουργεί πεπτιδικό δεσμό με την αμινομάδα του επόμενου αμινοξέος. Κατά την διαδικασία αυτή αποβάλλεται ένα μόριο νερού και γι αυτό πλέον οι μονάδες που αποτελούν την πολυπεπτιδική αλυσίδα ονομάζονται ***κατάλοιπα***. Τα διαφορετικά κατάλοιπα αποκτούν την διαφορετικότητά τους λόγο της πλευρικής ομάδας αφού είναι το μόνο τμήμα που μπορεί να διαφέρει στα κατάλοιπα, όπως και στα αμινοξέα, ενώ το

κεντρικό άτομο άνθρακα, η αμινομάδα (NH), η καρβονυλομάδα (C' =O) και το άτομο H είναι το τμήμα της κύριας αλυσίδας το οποίο είναι πανομοιότυπο σε όλα τα κατάλοιπα. Κατά την διαδικασία της σύγκρισης δύο πρωτεϊνών ως προς την δομή σημαντικό ρόλο για τον υπολογισμό της ομοιότητας έχουν κεντρικά άτομα άνθρακα.



Σχήμα 2.1.1.1: Δομή αμινοξέος.



Σχήμα 2.1.1.2: Σε αυτό το σχήμα αναπαρίσταται η ένωση δύο αμινοξέων, ο πεπτιδικός δεσμός που δημιουργείται από την ένωσή τους και το μόριο νερού που αποβάλλεται.

2.1.2 Τριτοταγής δομή πρωτεϊνών

Στα πρωτεϊνικά μόρια μπορούν να διακριθούν διάφορα δομικά επίπεδα οργάνωσης και τα οποία διακρίνονται στην πρωτοταγή, δευτεροταγή τριτοταγή και τεταρτοταγή δομή.

Η πρωτοταγής δομή αφορά την αλληλουχία των αμινοξέων στην πολυπεπτιδική αλυσίδα, είναι χαρακτηριστική για κάθε πρωτεΐνη και καθορίζεται από εντολές που δίνει το γενετικό υλικό. Η πρωτοταγής δομή μία πρωτεΐνης καθορίζει τις ανώτερες της δομές, δηλαδή το πώς θα αναδιπλωθεί και τη βιολογική της δράση. Όσο για την δευτεροταγή δομή αφορά τον τρόπο με τον οποίο συσπειρώνονται οι πολυπεπτιδικές αλυσίδες ώστε να αποκτήσουν μια τρισδιάστατη δομή. Η δευτεροταγής δομή των πρωτεϊνών προϋποθέτει την περιτύλιξη της πολυπεπτιδικής αλυσίδας σε μια ελικοειδή μορφή ή σε κάποιο άλλο κανονικό σχηματισμό. Ανάλογα με την αλληλουχία των αμινοξέων δημιουργείται α-έλικα ή β-πτυχωτή επιφάνεια. Σε κάποιες περιπτώσεις οι πρωτεΐνες αποτελούνται από δύο ή περισσότερες πολυπεπτιδικές αλυσίδες. Σε αυτή την περίπτωση συναντάται και η τεταρτοταγής δομή που περιγράφει τη συναρμολόγηση στο χώρο των αλυσίδων αυτών.

Η *τριτοταγής* δομή έχει μεγάλη σημαντικότητα διότι σε αυτήν οφείλεται η λειτουργία της πρωτεΐνης. Η τριτοταγής δομή δημιουργείται από την αναδίπλωση της πολυπεπτιδικής αλυσίδας πάνω στον εαυτό της και στο τελικό σχήμα που θα πάρει στο χώρο. Σε αυτή την δομή τμήματα του πολυπεπτιδίου που έχουν δευτερογενής διαμορφώσεις είτε ως α-έλικες, είτε ως β-πτυχωτές επιφάνειες, συνδεδεμένες με ενδιάμεσους πεπτιδικούς βρόγχους, αναδιπλώνονται σχηματίζοντας την τριτοταγή δομή της πρωτεΐνης. Αυτή η τρισδιάστατη διάταξη καθορίζεται από κάποιους παράγοντες οι οποίοι είναι οι υδρογονικοί δεσμοί των πλευρικών ομάδων των αμινοξέων σε γειτονικές περιελίξεις της πολυπεπτιδικής αλυσίδας, οι ιοντικές έλξεις μεταξύ θετικά και αρνητικά φορτισμένων ομάδων, οι υδρόφιλες αλληλεπιδράσεις και οι ομοιοπολικοί δεσμοί. Σε περίπτωση που η δομή αυτή επηρεαζόμενη από κάποιους παράγοντες αλλοιωθεί τότε επηρεάζονται οι βασικές φυσικές της ιδιότητες και η πρωτεΐνη χάνει την δράση της.

Λόγω του καθοριστικού ρόλου που έχει η τριτοταγής δομή στην λειτουργία της πρωτεΐνης, η σύγκριση πρωτεϊνών ως προς την δομή παρέχει επίγνωση σχετικά με την βιολογική λειτουργία μιας πρωτεΐνης. Όταν μία πρωτεΐνη της οποίας η λειτουργία είναι γνωστή εμφανίζει ομοιότητα, στην τριτοταγή της δομή, με μία

πρωτεΐνη με άγνωστη λειτουργία, η λειτουργία της δεύτερης πρωτεΐνης συμπεραίνεται από την λειτουργία της πρώτης. Επομένως η συσχέτιση μεταξύ των τριτοταγών δομών έχει μεγάλη σημασία σε μια εποχή όπου γίνεται μια έκρηξη της δομικής γονιδιωματικής με μία συνεχή αύξηση στον αριθμό των δομών με άγνωστες λειτουργίες.

2.1.3 Τα περιεχόμενα ενός PDB αρχείου.

Οι συγκρίσεις της τριτοταγούς δομής των πρωτεϊνών βασίζονται σε πληροφορίες σχετικές με την τρισδιάστατη διάταξη των εξεταζόμενων αλληλουχιών. Οι πληροφορίες αυτές μπορούν να αποκτηθούν με κρυσταλλογραφία ακτινών X ή φασματοσκοπία NMR. Μία πηγή τέτοιων πληροφοριών, που αφορούν τις τρισδιάστατες δομές των πρωτεϊνών, είναι τα *PDB* αρχεία, προερχόμενα από την ολοένα και μεγαλύτερη βάση της Protein Data Bank. Πρόκειται για αρχεία τα οποία έχουν κωδικοποιημένες ονομασίες, που σχετίζονται με την πρωτεΐνη στην οποία αναφέρονται, που αποτελούνται από τέσσερις χαρακτήρες και ο πρώτος χαρακτήρας αναπαριστά ένα ακέραιο αριθμό, ενώ οι υπόλοιποι τρεις μπορεί να αναπαριστούν και αυτοί έναν ακέραιο αριθμό, είτε ένα χαρακτήρα του αλφαβήτου. Τα αρχεία αυτά περιλαμβάνουν πληροφορίες που αφορούν τις συντεταγμένες των τριών διαστάσεων (X, Y, Z) του κάθε ατόμου στο χώρο, σε ένα ή και περισσότερα μόρια.

Κάθε γραμμή των PDB αρχείων έχει στην αρχή της μία “ετικέτα” για να διακρίνεται ο τύπος της πληροφορίας που περιέχει κάθε γραμμή. Οι “ετικέτες” αυτές δίνονται πιο κάτω με την σειρά που εμφανίζονται στο αρχείο:

- HEADER: περιλαμβάνει το όνομα του αρχείου και την ημερομηνία δημιουργίας του.
- COMPND: περιλαμβάνει το όνομα της πρωτεΐνης.
- SOURCE: αναφέρεται οργανισμός από τον οποίο πάρθηκε η πρωτεΐνη.
- AUTHOR: αναφέρονται τα άτομα που πρόσθεσαν τις πληροφορίες αυτού του αρχείου στην Protein Data Bank.
- REVDAT: αναγράφονται όλες οι ημερομηνίες αναθεώρησης των δεδομένων για αυτή την πρωτεΐνη.

- REMARK: περιλαμβάνονται αναφορές σε δημοσιεύσεις σχετικά με την δομή αυτής της πρωτεΐνης και γενικές πληροφορίες για τα περιεχόμενα του συγκεκριμένου αρχείου.
- SPRSDE: αναφέρονται παλαιότερα αρχεία συντεταγμένων που αφορούν την συγκεκριμένη δομή.
- SEQRES: δίνεται η αλληλουχία των αμινοξέων της πρωτεΐνης.
- FTNOTE: αναγράφονται διάφορες υποσημειώσεις.
- HET και FORMUL: περιλαμβάνονται οι συμπαράγωντες, οι προσθετικές ομάδες, οι αναστολείς και άλλες μη πρωτεϊνικές ουσίες οι οποίες υπάρχουν στην δομή.
- HELIX, SHEET, και TURN: αναφέρονται τα στοιχεία της δευτεροταγούς δομής της πρωτεΐνης.
- CRYST1, ORGI, και SCALE: δίνονται κάποιες γενικές πληροφορίες σχετικά με τις κρυσταλλικές πρωτεΐνες των οποίων ελήφθη η δομή με την μέθοδο της κρυσταλλογραφίας με ακτίνες Χ.
- ATOM και HETAM: περιλαμβάνονται οι συντεταγμένες των ατόμων που προσδιορίζουν την δομή της πρωτεΐνης.
- CONECT: περιλαμβάνονται οι δεσμοί που αφορούν δεσμούς με μη πρωτεϊνικά άτομα στο αρχείο.
- MASTER και END: σηματοδοτούν το τέλος του αρχείου.

Αριθμός ατόμου

Ταυτότητα κατάλοιπου

Αριθμός κατάλοιπου

Πεπτιδική αλυσίδα

Ταυτότητα ατόμου

Συντεταγμένες κάθε ατόμου για κάθε κατάλοιπο

						X	Y	Z			
ATOM	1290	N	ALA	B	24	19.190	9.112	6.083	1.00	0.00	N
ATOM	1291	CA	ALA	B	24	18.397	8.955	7.301	1.00	0.00	C
ATOM	1292	C	ALA	B	24	16.993	8.443	6.981	1.00	0.00	C
ATOM	1293	O	ALA	B	24	16.493	7.540	7.646	1.00	0.00	O
ATOM	1294	CB	ALA	B	24	18.321	10.274	8.058	1.00	0.00	C
ATOM	1295	H	ALA	B	24	19.517	10.003	5.838	1.00	0.00	H
ATOM	1296	HA	ALA	B	24	18.897	8.233	7.932	1.00	0.00	H
ATOM	1297	HB1	ALA	B	24	19.308	10.550	8.401	1.00	0.00	H
ATOM	1298	HB2	ALA	B	24	17.662	10.164	8.907	1.00	0.00	H
ATOM	1299	HB3	ALA	B	24	17.940	11.043	7.403	1.00	0.00	H
ATOM	1300	N	ARG	B	25	16.365	9.025	5.958	1.00	0.00	N
ATOM	1301	CA	ARG	B	25	15.018	8.626	5.546	1.00	0.00	C
ATOM	1302	C	ARG	B	25	14.951	7.131	5.221	1.00	0.00	C
ATOM	1303	O	ARG	B	25	14.112	6.410	5.763	1.00	0.00	O
ATOM	1304	CB	ARG	B	25	14.570	9.445	4.332	1.00	0.00	C

Εικόνα 2.1.3.1: Σε αυτή την εικόνα παρουσιάζονται κάποιες από τις γραμμές ενός PDB αρχείου που αφορούν την σημαντική πληροφορία για αυτή την εργασία,και που αφορά τις συντεταγμένες των ατόμων. Στην εικόνα σημειώνονται σε κόκκινο πλαίσιο οι γραμμές που αφορούν τα C_α άτομα οι συντεταγμένες των οποίων αφορούν τον αλγόριθμο που μελετάται σε αυτή την εργασία.

2.2 Διάφορες μέθοδοι σύγκρισης της δομής των πρωτεϊνών

Με την ολοένα αυξανόμενη σε μέγεθος βάση γνωστών τριτοταγών δομών των πρωτεϊνών και την μεγάλη σημασία της δομής των πρωτεϊνών, μεγαλώνει και η ανάγκη για σύγκριση μεταξύ των δομών. Αν και η τρισδιάστατη μορφή που αποκτά σύμφωνα με την τριτοταγή της δομή μία πρωτεΐνη καθορίζεται από την πρωτοταγή της δομή παρ' όλα αυτά μια σύγκριση των πρωτοταγών δομών δεν μπορεί να ανιχνεύσει ομοιότητα στην τριτοταγή δομή. Για το λόγο αυτό αναπτύχθηκαν διάφορες μέθοδοι που υλοποιούν την σύγκριση ως προς την τριτοταγή δομή.

Κάποιες από τις πιο αντιπροσωπευτικές μεθόδους, εκτός της μεθόδου CE η οποία θα μελετηθεί στο επόμενο κεφάλαιο, παρουσιάζονται πιο κάτω:

Dali

Η μέθοδος Dali, [11]και [13], χρησιμοποιεί τις τρισδιάστατες συντεταγμένες κάθε πρωτεΐνης για να υπολογίσει κατάλοιπο προς κατάλοιπο πίνακες αποστάσεων. Οι απομακρυσμένοι πίνακες αποσυντίθενται αρχικά σε στοιχειώδη αντίγραφα, για παράδειγμα σε υποπίνακες εξαπεπτίδιο-εξαπεπτίδιο. Στη συνέχεια παρόμοια αντίγραφα στους δύο πίνακες ζευγαρώνονται και συγκρίνονται με μεγαλύτερα συνεχή ζεύγη. Μια διαδικασία Monte Carlo χρησιμοποιείται για να βελτιστοποιηθεί ο βαθμός ομοιότητας ο οποίος εκφράζεται ως ενδομοριακές αποστάσεις. Πολλαπλές ευθυγραμμίσεις βελτιστοποιούνται παράλληλα, οδηγώντας σε ταυτόχρονη ανίχνευση της καλύτερης, της δεύτερης καλύτερης και της επόμενης κατά αυτή την διαδικασία λύσης. Η μέθοδος επιτρέπει την ευθυγράμμιση κενών οποιουδήποτε μήκους, αντιστροφή της κατεύθυνσης της αλυσίδας και ελεύθερη τοπολογική σύνδεση των ευθυγραμμισμένων τμημάτων. Η μέθοδος είναι πλήρως αυτοματοποιημένη και αναγνωρίζει δομικές ομοιότητες και κοινά δομικά βασικά τμήματα με ακρίβεια και ευαισθησία, ακόμα και κατά την παρουσία γεωμετρικών παραμορφώσεων.

CTSS

Η μέθοδος αυτή στηρίζεται στη θεωρία της διαφορικής γεωμετρίας με την σύγκριση καμπών στον τρισδιάστατο χώρο [12]. Παράγονται οι υπογραφές σχήματος για πρωτεΐνες οι οποίες είναι σταθερές, εντοπισμένες, εύρωστες, συμπυκνωμένες και που έχουν βιολογική σημασία. Για να βελτιωθεί η ακρίβεια του ταιριάσματος εξομαλύνονται τα ακατέργαστα και με θόρυβο δεδομένα συντεταγμένων με την χρήση ενός εύκαμπτου κανόνα για χάραξη μεγάλων καμπυλών. Για την βελτίωση της αποδοτικότητας υιοθετείται η στρατηγική “χονδροειδές προς λεπτό”. Χρησιμοποιείται μία αποδοτική τεχνική κατακερματισμού με την οποία αποκλείονται οι απίθανοι υποψήφιοι, ώστε να εκτελεστούν οι λεπτομερείς κατά ζεύγος ευθυγραμμίσεις μόνο για έναν μικρό αριθμό υποψηφίων που επιζούν της διαδικασίας διαλογής.

PADS

Η PADS [15] είναι μια νέα προσέγγιση ανάσυρσης δεδομένων για την αναζήτηση ομοιότητας και την ανακάλυψη γνώσης στις πρωτεϊνικές βάσεις δεδομένων δομών. Το PADS (Protein structure Alignment by Directional shape Signatures) ενσωματώνει τις τρισδιάστατες συντεταγμένες των κύριων ατόμων κάθε αμινοξέος και εξάγει μια γεωμετρική υπογραφή μορφής μαζί με την κατεύθυνση κάθε αμινοξέος. Κατά συνέπεια, κάθε πρωτεϊνική δομή παρουσιάζεται από μια σειρά πολυδιάστατων διανυσμάτων χαρακτηριστικών γνωρισμάτων που αντιπροσωπεύουν την τοπική γεωμετρία, τη μορφή, την κατεύθυνση, και τις βιολογικές ιδιότητες των μορίων αμινοξέος της. Επιπλέον, ένας πίνακας αποστάσεων υπολογίζεται και ενσωματώνεται σε έναν τοπικό δυναμικό αλγόριθμο προγραμματισμού ευθυγράμμισης για να βρει τα παρόμοια τμήματα δύο πρωτεϊνικών δομών που ακολουθούνται από ένα βήμα ευθυγράμμισης ακολουθίας για ένα αποδοτικότερο φιλτράρισμα. Η βέλτιστη επαλληλία των ανιχνευμένων παρόμοιων περιοχών χρησιμοποιείται για να αξιολογηθεί η ποιότητα των αποτελεσμάτων. Ο προτεινόμενος αλγόριθμος είναι γρήγορος και ακριβής και ως εκ τούτου θα μπορούσε να χρησιμοποιηθεί για την ανακάλυψη, ανάλυση και γνώση των μεγάλων πρωτεϊνικών δομών.

TOPOFIT

Στην μέθοδο TOPOFIT [16] η ομοιότητα των πρωτεϊνικών δομών αναλύεται χρησιμοποιώντας τρισδιάστατα τριγωνομετρικά μοντέλα Delaunay προερχόμενα από την πρωτεϊνική δομή. Έχει διαπιστωθεί ότι οι πρωτεΐνες που σχετίζονται ως προς την δομή έχουν ένα κοινό χωρικά αμετάβλητο μέρος, ένα σύνολο τετραέδρων, τα οποία περιγράφονται από μαθηματική άποψη ως κοινός χωρικός παραδιαγραμματικός όγκος της τρισδιάστατης γραφικής παράστασης επαφών που προέρχεται από το Delaunay tessellation (DT). Με βάση αυτήν την ιδιότητα των πρωτεϊνικών δομών παρουσιάζεται μια νέα κοινή μέθοδος υπέρθεσης του όγκου (TOPOFIT) για να παραχθούν οι δομικές ευθυγραμμίσεις των πρωτεϊνών. Η υπέρθεση των σχεδίων της DT επιτρέπει σε κάποιον να προσδιορίσει μεμονωμένα έναν κοινό αριθμό ισοδύναμων κατάλοιπων στη δομική ευθυγράμμιση, με άλλα

λόγια, το TOPOFIT προσδιορίζει ένα σημείο χαρακτηριστικών γνωρισμάτων στην RMSD (Ne) καμπύλη, ένα σημείο topomax, μέχρι το οποίο δύο δομές αντιστοιχούν η μια στην άλλη συμπεριλαμβανομένων των επαφών των πρωτεϊνικών σοπνδυλικών στηλών και των μεταξύ των καταλοίπων επαφών, ενώ ο αυξανόμενος αριθμός κακών συνδυασμών μεταξύ των σχεδίων της DT εμφανίζεται σε μεγαλύτερο RMSD (NE) μετά από το σημείο topomax. Το σημείο topomax είναι παρόν σε όλες τις ευθυγραμμίσεις από τις διάφορες πρωτεϊνικές δομικές κατηγορίες επομένως, η μέθοδος TOPOFIT προσδιορίζει τα κοινά, αμετάβλητα δομικά μέρη μεταξύ των πρωτεϊνών. Η μέθοδος TOPOFIT προσθέτει τις νέες ευκαιρίες για τη συγκριτική ανάλυση των πρωτεϊνικών δομών και για τις πιο λεπτομερείς μελέτες για την κατανόηση των μοριακών αρχών της τριτοταγούς οργάνωσης και της λειτουργίας των πρωτεϊνών.

FAST

Η μέθοδος FAST [17] χρησιμοποιεί ένα κατευθυντικά-βασισμένο σχέδιο βαθμολόγησης των αποτελεσμάτων για να συγκριθούν οι ενδομοριακές σχέσεις κατάλοιπο προς κατάλοιπο σε δύο δομές. Χρησιμοποιεί μια ευρεστική μέθοδο μέσω απόκλισης για να προωθήσει τη σποραδικότητα στη γραφική παράσταση κατάλοιπου-ζευγαριού και να διευκολύνει την ανίχνευση του βέλτιστου. Προκειμένου να εξεταστεί η γενική ακρίβεια του FAST, καθορίζεται η ευαισθησία και η ιδιομορφία του με την ταξινόμηση SCOP (έκδοση 1.61) ως χρυσός κανόνας.

Παρ’ όλες αυτές τις μεθόδους σε αυτή την εργασία επιλέχθηκε ο αλγόριθμος CE, που είναι μία γρήγορη και ακριβής μέθοδος, που πετυχαίνει μία σημαντική μείωση του εύρους του πεδίου αναζήτησης και η λειτουργία της βασίζεται σε ευριστικές μεθόδους. Εκτός από αυτά μεγάλο ρόλο στην επιλογή του έπαιξε και το γεγονός ότι περιλαμβάνεται στο πακέτο-σημείο αναφοράς του Bioperf.

3

Ο Αλγόριθμος CE

Στο κεφάλαιο αυτό περιγράφεται ο αλγόριθμός Combinatorial Extension (CE) και γίνεται αναφορά στην μοντελοποίηση που έγινε γι αυτό τον αλγόριθμο στην γλώσσα προγραμματισμού C.

3.1 Μέθοδος Combinatorial Extension (CE)

Η CE είναι μια μέθοδος υπολογισμού της δομικής ομοιότητας των πρωτεϊνών κατά την οποία συγκρίνονται δύο οντότητες κατά ζεύγη για να κριθεί ποιό από το κάθε ζεύγος είναι προτιμότερο σύμφωνα με τα κριτήρια τα οποία έχουν τεθεί. Η CE συγκρίνει δύο πολυπεπτιδικές αλυσίδες χρησιμοποιώντας χαρακτηριστικά της τοπικής τους γεωμετρίας όπως αυτή ορίζεται από τα ανύσματα των θέσεων των C_a ατόμων.

Ο αλγόριθμος CE, ο οποίος δημοσιεύτηκε από τους Ilya N. Shindyalov και Philip E. Bourne [1], εμπλέκει την μέθοδο CE για την εύρεση ενός μονοπατιού ομοιότητας μεταξύ δύο πολυπεπτιδικών αλυσίδων αποτελούμενο από AFPs (aligned fragment pairs). Τα AFPs είναι ζεύγη αποτελούμενα από τμήματα των δύο πρωτεϊνών, ένα από κάθε πρωτεΐνη. Τα τμήματα αυτά έχουν καθορισμένο μήκος και για να είναι δυνατή η χρήση τους πρέπει να παρέχουν μία δομική ομοιότητα η οποία βασίζεται στην τοπική γεωμετρία. Συνδυασμοί από AFPs οι οποίοι μπορεί να αποτελέσουν συνεχή μονοπάτια ομοιότητας επιλεκτικά επεκτείνονται ή απορρίπτονται με σκοπό να βρεθεί το βέλτιστο μονοπάτι ομοιότητας.

Ο αλγόριθμος CE έχει μια σημαντική μείωση στον εύρος του πεδίου αναζήτησης και εμπειρικά καθορίζει μία λειτουργία χρησιμοποιώντας ευριστικές μεθόδους. Η λειτουργία αυτή υποθέτει πρώτον ότι το μονοπάτι είναι συνεχές όταν περιλαμβάνει κενά και δεύτερον ότι υπάρχει ένα βέλτιστο μονοπάτι.

Διάφορες πρωτεϊνικές ιδιότητες μπορούν να χρησιμοποιηθούν για να υπολογιστεί η δομική τους ομοιότητα και πρόσφατα ένα ολοκληρωμένο σύνολο ιδιοτήτων παρουσιάστηκε και ελέγχθηκε για την σύγκριση ως προς την δομή. Αρκετές από αυτές τις ιδιότητες μπορούν να χρησιμοποιηθούν από τον αλγόριθμο CE όπως είναι και η απόσταση μεταξύ των C_a ατόμων των κατάλοιπων που δομούν μία πολυπεπτιδική αλυσίδα (inter-residue distance).

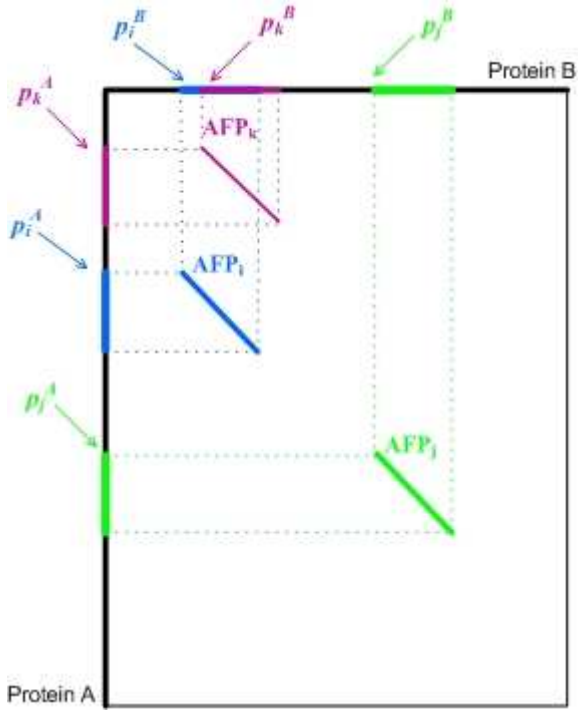
Η μέθοδος αυτή έχει ελεγχτεί συγκρινόμενη με τις μεθόδους Dali και VAST χρησιμοποιώντας ένα αντιπροσωπευτικό δείγμα από πολυπεπτιδικές αλυσίδες με δομικές ομοιότητες.

3.2 Ανάλυση του αλγορίθμου Combinatorial Extension (CE)

Σύμφωνα με τον αλγόριθμο CE το αποτέλεσμα της σύγκρισης δύο πρωτεϊνικών δομών *A* και *B*, μήκους n^A και n^B αντίστοιχα, είναι το μεγαλύτερο σε μήκος συνεχές μονοπάτι *P*, αποτελούμενο από AFPs με μέγεθος *m*, σε ένα πίνακα ομοιότητας *S*, μεγέθους $(n^A-m)*(n^B-m)$. Ο πίνακας αυτός αντιπροσωπεύει όλα τα AFPs οποία πληρούν τα κριτήρια ομοιότητας και μπορούν να αποτελέσουν μέρος ενός μονοπατιού ομοιότητας.

Πριν ακολουθήσει η περιγραφή των βημάτων του αλγορίθμου είναι απαραίτητο για την καλύτερη κατανόησή τους να καθοριστεί με περισσότερη σαφήνεια η έννοια του AFP. Όπως εννοήθηκε και στην προηγούμενη παράγραφο δύο πρωτεΐνες *A* και *B* μεγέθους n^A και n^B αντίστοιχα, παίρνουμε $(n^A-m)*(n^B-m)$ AFPs τα οποία όμως για να μπορέσουν να προστεθούν σε ένα μονοπάτι δομικής ομοιότητας θα πρέπει να πληρούν κάποια κριτήρια ομοιότητας τα οποία θα αναλυθούν, αφού δοθούν κάποιες λεπτομέρειες σχετικά με τα AFPs. Το μέγεθος *m* ενός AFP είναι σταθερό για το οποίο μια λογική τιμή, η οποία επιλέγεται ως συνήθως, είναι το 8 και θα θεωρείται σε αυτή την εργασία ως η προκαθορισμένη τιμή για το μέγεθος *m*. Οπόταν ένα AFP είναι η αντιστοιχία m διαδοχικών κατάλοιπων της μιας πρωτεΐνης, με m διαδοχικά κατάλοιπα της άλλης πρωτεΐνης. Για τον σχηματισμό όλων των AFPs που μας δίνουν οι δύο πρωτεΐνες πρέπει κάθε m-άδα από την μία πρωτεΐνη να σχηματίσει ζεύγη με όλες τις m-άδες από την άλλη πρωτεΐνη. Δηλαδή κάνοντας την παραδοχή ότι το *m* είναι 8 και παρουσιάζοντας την ακολουθία από τα κατάλοιπα που αποτελούν την πρωτεΐνη σαν ένα διάνυσμα, τότε ο τρόπος με τον οποίο χωρίζεται μια

πρωτεΐνη σε m-άδες είναι: από 1-8 η πρώτη m-άδα, από 2-9 η δεύτερη, και συνεχίζοντας έτσι η τελευταία m-άδα αποτελείται από τα (n^A-(m-1)) έως και n^A. Επίσης ακόμα μία πληροφορία η οποία χαρακτηρίζει ένα AFPi (όπου i από 1 μέχρι *(n^A-m)*(n^B-m)*) είναι η πληροφορία για την αρχική θέση p_i^A της m-άδας στην πρωτεΐνη A και την αρχική θέση p_i^B της m-άδας στην πρωτεΐνη B.



Σχήμα 3.2.1: Σε αυτό το σχήμα παρατηρούνται τρία AFPs. Για τον σχηματισμό του κάθε AFP χρειάζεται ένα τμήμα σταθερού μεγέθους από κάθε πρωτεΐνη και κάθε AFP χαρακτηρίζεται με κάποιο ξεχωριστό δείκτη. Παρ' όλα αυτά όπως παρατηρείται και στο σχήμα ο μοναδικός τρόπος προσδιορισμού του κάθε AFP είναι το ζεύγος των αρχικών θέσεων, των τμημάτων που το αποτελούν, στην κάθε μία πρωτεΐνη.

Το κριτήριο με το οποίο ελέγχεται ένα AFP αν μπορεί να αποτελέσει μέρος ενός μονοπατιού δίνεται από την σχέση 3.2.1.

$$Dnn < D_0 \qquad (3.2.1)$$

Στην πιο πάνω σχέση το D_o είναι ένα κατώφλι ομοιότητας και όταν αφορά τον έλεγχο ενός υποψήφιου AFP όπως σε αυτή την περίπτωση έχει την χαρακτηριστική τιμή D_o=3Å. Το D_{ij} είναι η απόσταση μεταξύ των τμημάτων που απαρτίζουν τα AFPs i (AFPi) και j (AFPj). Στην σχέση 3.2.1 παρατηρούμε ότι ή απόσταση αυτή συμβολίζεται ως D_{nn}. Αυτό συμβαίνει διότι εξετάζεται αν το AFPn τηρεί το κριτήριο ομοιότητας για να μπορεί να αποτελέσει ένα AFP ικανό να λάβει μέρος στον σχηματισμό ενός μονοπατιού, άρα ο υπολογισμός της απόστασης γίνεται μόνο εντός του τμήματος που περιλαμβάνει το AFPn και έτσι τα i και j έχουν την κοινή τιμή n.

Ο υπολογισμός του D_{ij} μπορεί να γίνει με τους εξής δύο τρόπους.

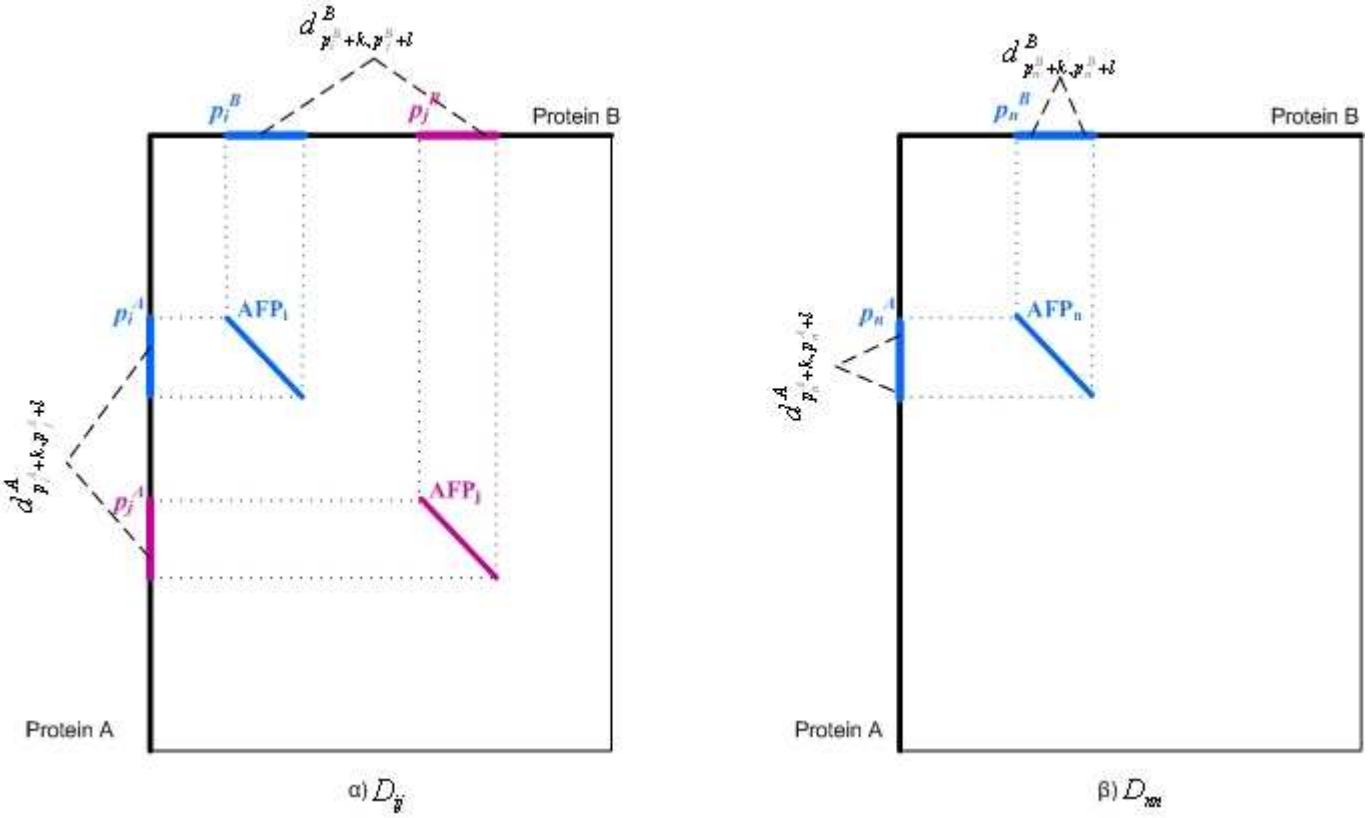
$$D_{ij} = \frac{1}{m}\left(\left| d^A_{p_i^A p_j^A} - d^B_{p_i^B p_j^B} \right| + \left| d^A_{p_i^A+m-1, p_j^A+m-1} - d^B_{p_i^B+m-1, p_j^B+m-1} \right| + \sum_{k=1}^{m-2} \left| d^A_{p_i^A+k, p_j^A+m-1-k} - d^B_{p_i^B+k, p_j^B+m-1-k} \right| \right) \quad (3.2.2)$$

Με τον τρόπο αυτόν η απόσταση D_{ij} υπολογίζεται χρησιμοποιώντας ένα "ανεξάρτητο" σετ αποστάσεων μεταξύ των κατάλοιπων, όπου κάθε κατάλοιπο συμμετέχει μόνο μία φορά σε αυτό το σετ. Όσον αφορά το d^A_{ij}, που παρατηρείται στην σχέση 3.2.2, είναι η απόσταση μεταξύ των κατάλοιπων i και j της πρωτεΐνης, βασισμένη στις συντεταγμένες, X, Y, Z, των C_a ατόμων. Το ίδιο ισχύει και για το d^B_{ij}. Η απόσταση αυτή υπολογίζεται ως η ευκλείδεια απόσταση μεταξύ των συντεταγμένων των C_a ατόμων των κατάλοιπων που δομούν την πρωτεΐνη. Αυτή η μέθοδος υπολογισμού της απόστασης D_{ij} χρησιμοποιείται κυρίως για να αξιολογηθεί κατά πόσο δύο AFPs, ένα που ήδη υπάρχει στο μονοπάτι και ένα που πρόκειται να προστεθεί, συνδυάζονται μεταξύ τους.

Εναλλακτικά η απόσταση D_{ij} υπολογίζεται χρησιμοποιώντας ένα πλήρες σετ αποστάσεων μεταξύ των κατάλοιπων, όπου λαμβάνονται υπόψιν όλες οι δυνατές αποστάσεις εκτός από των γειτονικών κατάλοιπων. Ο υπολογισμός αυτός δίνεται από τον τύπο 3.2.3. Η μέθοδος αυτή χρησιμοποιείται για την αξιολόγηση ενός AFP, δηλαδή κατά πόσο δύο τμήματα, ένα από κάθε πρωτεΐνη, τα οποία σχηματίζουν ένα AFP ταιριάζουν μεταξύ τους.

$$Dij = \frac{1}{m^2}\left(\sum_{k=0}^{m-1} \sum_{l=0}^{m-1} \left| d^A_{p_i^A+k, p_j^A+l} - d^B_{p_i^B+k, p_j^B+l} \right| \right) \quad (3.2.3)$$

Σύμφωνα με την ανάλυση που δόθηκε για την χρήση αυτών των δύο σχέσεων για τον υπολογισμό του D_{nn} στην σχέση 3.2.1 χρησιμοποιείται η 3.2.3.



Σχήμα 3.2.2: α) Αυτό το σχήμα αφορά τον υπολογισμό του D_{ij} και παρατηρείται ότι οι ευκλείδειες αποστάσεις *d* υπολογίζονται μεταξύ δύο διαφορετικών AFPs, των AFPi και AFPj.

β) Αντίθετα σε αυτό το σχήμα που αναφέρεται στον υπολογισμό του $D_{nn,}$ οι ευκλείδειες αποστάσεις *d* υπολογίζονται μεταξύ των κατάλοιπων του ίδιου AFP, του AFPn.

Μετά την αναλυτική επεξήγηση του όρου AFP, και ό,τι αφορά τις ιδιότητες αυτού, μπορεί να ακολουθήσει η περιγραφή της δημιουργίας ενός μονοπατιού στον πίνακα ομοιότητας S.

Καταρχάς σύμφωνα με τον αλγόριθμό κάθε AFP μπορεί να αποτελέσει την αρχή ενός μονοπατιού. Επομένως κατά την διαδικασία αναζήτησης του βέλτιστου μονοπατιού θα δημιουργηθούν τόσα μονοπάτια, όσα είναι και τα AFPs στον πίνακα ομοιότητας τα οποία ικανοποιούν την σχέση 3.2.1. Αφού λοιπόν ένα AFP ικανοποιεί

τους πιο πάνω περιορισμούς ακολουθεί η διαδικασία επέκτασης αυτού του μονοπατιού. Πριν γίνει όμως αυτό, με την σχέση 3.2.4 εξετάζεται αν το μονοπάτι θα επεκταθεί περαιτέρω ή θα τερματιστεί.

$$\frac{1}{n^2}\sum_{i=0}^{n}\sum_{j=0}^{n} D_{ij} < D_1 \qquad (3.2.4)$$

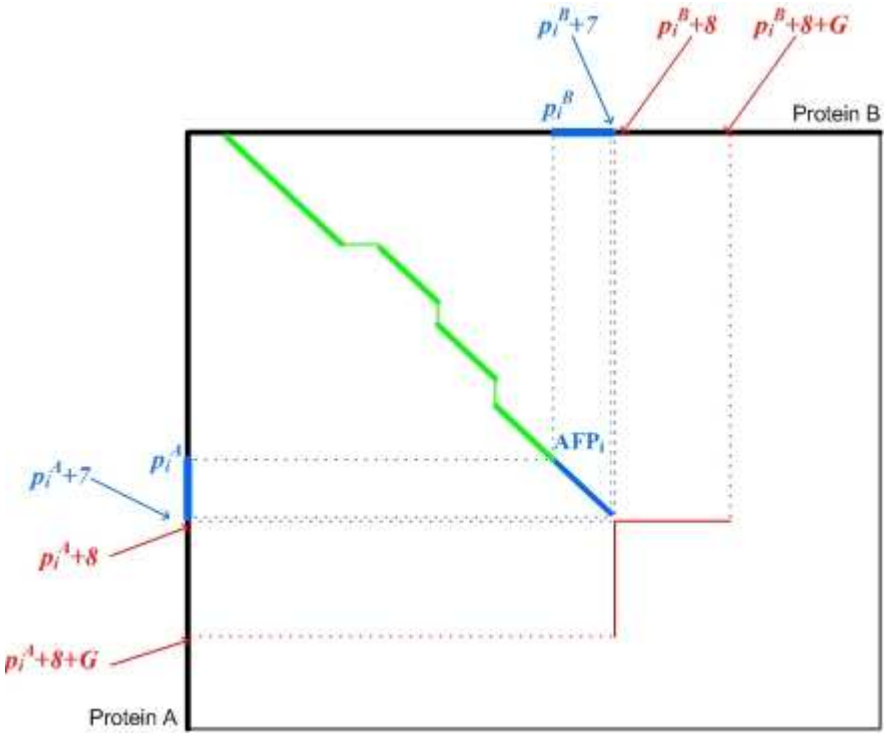
Εδώ το κατώφλι ομοιότητας αντιπροσωπεύεται από το D_1 και έχει την τιμή D_1=4Å, ενώ ο υπολογισμός του D_{ij} γίνεται με την χρήση της 3.2.2. Το n είναι ο αριθμός του AFP που πρόκειται να προστεθεί στο μονοπάτι, δηλαδή το μέγεθος του μονοπατιού συν 1.

Αν το μονοπάτι μπορεί να επεκταθεί περαιτέρω το επόμενο AFP που μπορεί να προστεθεί στο μονοπάτι είναι το AFP για το οποίο αν θεωρηθεί p_i^A η αρχική θέση της m-άδας στην πρωτεΐνη A και p_i^B η αρχική θέση της m-άδας στην πρωτεΐνη B για το αρχικό AFP, και p_{i+1}^A και p_{i+1}^B οι αρχικές θέσεις για το AFP το οποίο πρόκειται να προστεθεί στο μονοπάτι, οι αρχικές του θέσεις είναι οι $p_{i+1}^A = p_i^A + m$ και $p_{i+1}^B = p_i^B + m$. Υπάρχει και η δυνατότητα όμως να προστεθεί ένα από τα AFPs που πληροί τις εξής προϋποθέσεις: είτε $p_{i+1}^A > p_i^A + m$ και $p_{i+1}^B = p_i^B + m$, είτε $p_{i+1}^A = p_i^A + m$ και $p_{i+1}^B > p_i^B + m$. Δηλαδή μπορούν να υπάρξουν κενές θέσεις μεταξύ δύο διαδοχικών AFPs μόνο σε μια από τις δύο πρωτεΐνες. Το πόσα κενά μπορούμε να υπάρχουν περιορίζεται από τις σχέσεις 3.2.5 και 3.2.6.

$$p_{i+1}^A \le p_i^A + m + G \qquad (3.2.5)$$

$$p_{i+1}^B \le p_i^B + m + G \qquad (3.2.6)$$

Εμπειρικά αποδείχτηκε ότι μία καλή τιμή για το G είναι 30. Δηλαδή μπορούν να υπάρξουν μέχρι 30 κενά.



Σχήμα 3.2.3: Σε αυτό το σχήμα παρουσιάζεται με πράσινο χρώμα ένα μονοπάτι και με μπλε χρώμα είναι το τελευταίο AFP του μονοπατιού, το AFPi το οποίο έχει ήδη προστεθεί στο μονοπάτι. Με μπλε χρώμα δίνονται επίσης και τα όρια του AFPi στις δύο πρωτεΐνες. Οι κόκκινες συνεχής γραμμές που υπάρχουν στο σχήμα καθώς και το σημείο όπου αυτές τέμνονται υποδηλώνουν τις θέσεις στις από τις οποίες μπορεί να ξεκινάει το επόμενο AFP που θα προστεθεί στο μονοπάτι.

Παρατηρούμε λοιπόν ότι έχουμε πολλές, και συγκεκριμένα 61, επιλογές σχετικά με το ποιο AFP θα προστεθεί στο μονοπάτι. Η επιλογή για το πιο από όλα αυτά τα AFPs θα προστεθεί στο μονοπάτι γίνεται σύμφωνα με την σχέση 3.2.7.

$$\frac{1}{n-1}\sum_{i=0}^{n-1} D_{in} < D_1 \qquad (3.2.7)$$

Σύμφωνα με την μέθοδο θα προστεθεί στο μονοπάτι το AFP για το οποίο υπολογίζεται η μικρότερη τιμή, σε σχέση με τα υπόλοιπα, όπως αυτή υπολογίζεται από το αριστερό μέλος της 3.2.7 και ικανοποιεί και την ανισότητα. Μια ακόμα απαραίτητη προϋπόθεση είναι το AFP να ικανοποιεί την 3.2.1. Επίσης στην

περίπτωση αυτή της σχέσης 3.2.7 το κατώφλι ομοιότητας αντιπροσωπεύεται από το D_1=4Å, ενώ ο υπολογισμός του D_{ij} γίνεται με την χρήση της 3.2.2.

Αν κανένα από τα 61 υποψήφια, για να προστεθούν στο μονοπάτι AFPs, δεν πληροί έναν από τους περιορισμούς που υποδηλώνουν οι σχέσεις 3.2.1 και 3.2.7 τότε τερματίζεται η διαδικασία επέκτασης του μονοπατιού και αρχίζει η διαδικασία για το επόμενο. Αλλιώς ο τερματισμός του μονοπατιού θα έρθει είτε λόγο της σχέσης 3.2.4, ή λόγο του ότι το μονοπάτι έχει φτάσει στα όρια των μεγεθών των δύο πρωτεϊνών και έτσι δεν υπάρχει πλέον άλλο AFP για να προστεθεί στο μονοπάτι.

Στο τέλος όλης της διαδικασίας μπορεί να υπάρχουν περισσότερα του ενός μονοπάτια με το βέλτιστο μήκος. Έτσι για τον λόγο αυτό υπολογίζεται και μία άλλη τιμή για κάθε μονοπάτι, η τιμή RMSD, και το μονοπάτι που συνδυάζει το μεγαλύτερο μήκος και τον μικρότερο RMSD επιλέγεται ως το βέλτιστο.

3.3 Μελέτη και μοντελοποίηση του αλγορίθμου CE με την χρήση της γλώσσας προγραμματισμού C

Κατά την μελέτη του αλγορίθμου *CE* μοντελοποιήθηκε με την χρήση της γλώσσας προγραμματισμού C με στόχο τον έλεγχο του βαθμού κατανόησής του, αλλά και την χρήση της διαδικασίας αυτής στην συνέχεια στην φάση της υλοποίησης του αλγορίθμου σε αναδιατασσόμενη λογική, χρησιμοποιώντας την ως βάση κατά την σχεδιαστική διαδικασία. Ο έλεγχος του κώδικα της γλώσσας C έγινε σε σύγκριση με τα αποτελέσματα τις υλοποίησης CE που περιλαμβάνεται στο πακέτο Bioperf και είναι μια υλοποίηση σε γλώσσα προγραμματισμού C++, αλλά και με ένα μέρος των αποτελεσμάτων από την δημοσίευση με τίτλο *Protein Structure Alignment by Incremental Combinatorial Extension (CE) of the Optimal Path* των Ilya N. Shindyalov and Philip E. Bourne, για τα οποία έχουν ληφθεί υπόψη οι ίδιες συνθήκες και παράμετροι κάτω από τις οποίες έγινε η υλοποίηση σε C.

Λόγω του ότι κατά την μελέτη των αποτελεσμάτων παρατηρήθηκε ότι η υλοποίηση που έγινε στην μοντελοποίηση για την 3.2.4 δεν είχε τα αναμενόμενα αποτελέσματα όπως αυτά ήταν εμφανή στην υλοποίηση του Bioperf, αφού ελέγχθηκαν αναλυτικά όλες οι υποσυναρτήσεις της συνάρτησης που παρατηρήθηκε η διαφορά και δεν παρατηρήθηκε κάποιο σφάλμα σε αυτές αποφασίστηκε ότι δεν θα χρησιμοποιηθεί αυτή η συνάρτηση αφού η πολυπλοκότητα με την οποία έγινε η

υλοποίηση σε C++ δεν βοήθησε στην αποσφαλμάτωση του κώδικα μοντελοποίησης. Για τον λόγο αυτό, αλλά και λόγο έλλειψης πληροφοριών σχετικά με τον υπολογισμό του RMSD στην δημοσίευση που αναφέρθηκε και προηγουμένως, από την οποία μελετήθηκαν τα βήματα του αλγορίθμου, παραλήφθηκε και το μέρος αυτό. Έτσι αποφασίστηκε ότι ως έξοδος στον κώδικα μοντελοποίησης θα δίνεται το βέλτιστο μέγεθος μονοπατιού κάτι που μπορεί να ελεγχθεί και με την υλοποίηση που περιλαμβάνεται στο Bioperf αλλά και με την δημοσίευση του αλγορίθμου, αφού και σε αυτό αναφέρονται τα αποτελέσματα με αυτό τον τρόπο για κάποιες περιπτώσεις χωρίς την χρήση της συνάρτησης υπολογισμού του RMSD.

Λαμβάνοντας υπόψη τα προβλήματα που παρουσιάστηκαν και τα οποία αναφέρθηκαν στην προηγούμενη παράγραφο μοντελοποιήθηκε η διαδικασία δημιουργίας όλων των μονοπατιών χωρίς όμως την χρήση του 3.2.4 η οποία θα ήταν χρήσιμη για τον τερματισμό ενός μονοπατιού νωρίτερα απ’ ότι θα γινόταν χωρίς την χρήση του παρατηρώντας ένα βαθμό συνδυασμού των AFPs που ήδη έχουν προστεθεί στο μονοπάτι μικρότερα από τον επιθυμητό. Παρ’ όλα αυτά μετά τον έλεγχο αρκετών περιπτώσεων παρατηρήθηκε ότι και χωρίς την χρήση αυτής της σχέσης τα αποτελέσματα συνέπιπταν με τα επιθυμητά.

Επίσης κατά την διαδικασία επιβεβαίωσης της ορθότητας της 3.2.3 που χρησιμοποιείται για τον υπολογισμό του D_{nn}, μετά από έλεγχο που έγινε παρεμβάλλοντας κώδικα ελέγχου σε πολλά σημεία της υλοποίησης σε C++ του Bioperf παρουσιάστηκε μια διαφορά μεταξύ των αποτελεσμάτων του κώδικα που υλοποιήθηκε κατά την μοντελοποίηση. Έτσι αφού επιβεβαιώθηκε ότι η υλοποίηση της 3.2.3 υλοποιήθηκε ορθά κατά την μοντελοποίηση ακολούθησε αναλυτική μελέτη της υλοποίησης σε C++ του Biopeft και βρέθηκε ότι σε αυτή στην υλοποίηση στην περίπτωση του D_{nn}, το D_{ij} υπολογίζεται μέσω της σχέσης 3.3.1. Επομένως για να μπορεί να γίνει η σύγκριση των αποτελεσμάτων μεταξύ των δύο εφαρμογών για την διαδικασία την μοντελοποίησης, αλλά και στην συνέχεια κατά την διαδικασία της υλοποίησης σε αναδιατασσόμενη λογική στην περίπτωση του D_{nn}, γίνεται χρήση της 3.3.1 για τον υπολογισμό του D_{ij}.

$$Dij = \frac{1}{21}\left(\sum_{k=0}^{m-2}\sum_{l=k+2}^{m}\left|d^{A}_{p_i^A+k,\,p_j^A+l} - d^{B}_{p_i^B+k,\,p_j^B+l}\right|\right) \qquad (3.3.1)$$

Ως είσοδοι χρησιμοποιήθηκαν PDB αρχεία που χρησιμοποιήθηκαν και ως είσοδοι στη εφαρμογή του Bioperf. Κατά το διάβασμα αυτών των αρχείων επιλέγεται η χρήσιμη πληροφορία που μας ενδιαφέρει και η οποία αναφέρεται στις συντεταγμένες X, Y, Z, των C_a ατόμων κάθε πρωτεΐνης. Η πληροφορία αυτή αποθηκεύεται σε δύο πίνακες, όπου ο κάθε πίνακας αντιστοιχεί σε μία από τις δύο πρωτεΐνες. Στην συνέχεια με την χρήση της 3.2.1 δημιουργείται ένας πίνακας με τις διαστάσεις του πίνακα *S* και του οποίου κάθε στοιχείο του αντιπροσωπεύει ένα AFP. Έτσι οι γραμμές αυτού του πίνακα αναφέρονται σε θέσεις στην πρωτεΐνη *A*, ενώ οι στήλες σε θέσεις στην πρωτεΐνη *B*. Ο συνδυασμός μίας γραμμής και μίας στήλης αντιπροσωπεύουν ένα AFP και συγκεκριμένα την αρχική θέση p_i^A της m-άδας στην πρωτεΐνη A και την αρχική θέση p_i^B της m-άδας στην πρωτεΐνη B αντίστοιχα. Εδώ πρέπει να αναφερθεί ότι σε όλη την διαδικασία η αναφορά σε κάποιο AFP γίνεται χρησιμοποιώντας τις τιμές p_i^A και p_i^B, που του αντιστοιχούν. Έτσί στην θέση αυτή του πίνακα αποθηκεύεται η τιμή 1 στην περίπτωση που ισχύει η ανισότητα της 3.2.1 για το συγκεκριμένο AFP και 0 στην περίπτωση που δεν ισχύει. Με αυτό τον τρόπο, στην συνέχεια κατά την δημιουργία των μονοπατιών, όταν πρέπει να ελεγχθεί κατά πόσο τα δύο τμήματα που απαρτίζουν το AFP ταιριάζουν μεταξύ τους στο βαθμό ώστε να μπορούν να αποτελέσουν μέρος ενός μονοπατιού, όπως αυτό αποφασίζεται από την 3.2.1, το μόνο που πρέπει να γίνει είναι να προσπελαστεί η θέση του πίνακα που αντιστοιχεί στο συγκεκριμένο AFP. Έτσι αποφεύγεται η διαδικασία υπολογισμού του D_{nn} για το AFPn περισσότερες από μία φορές. Αφού γίνει αυτή η διαδικασία με την χρήση αυτού του πίνακα , της σχέσης 3.2.7 και λαμβάνοντας υπόψη τους περιορισμούς και την διαδικασία όπως περιγράφεται στην προηγούμενη ενότητα δημιουργούνται όλα τα μονοπάτια και υπολογίζεται το βέλτιστο μέγεθος μονοπατιού.

4

Σχεδίαση και Υλοποίηση της Αρχιτεκτονικής για τον Αλγόριθμο CE

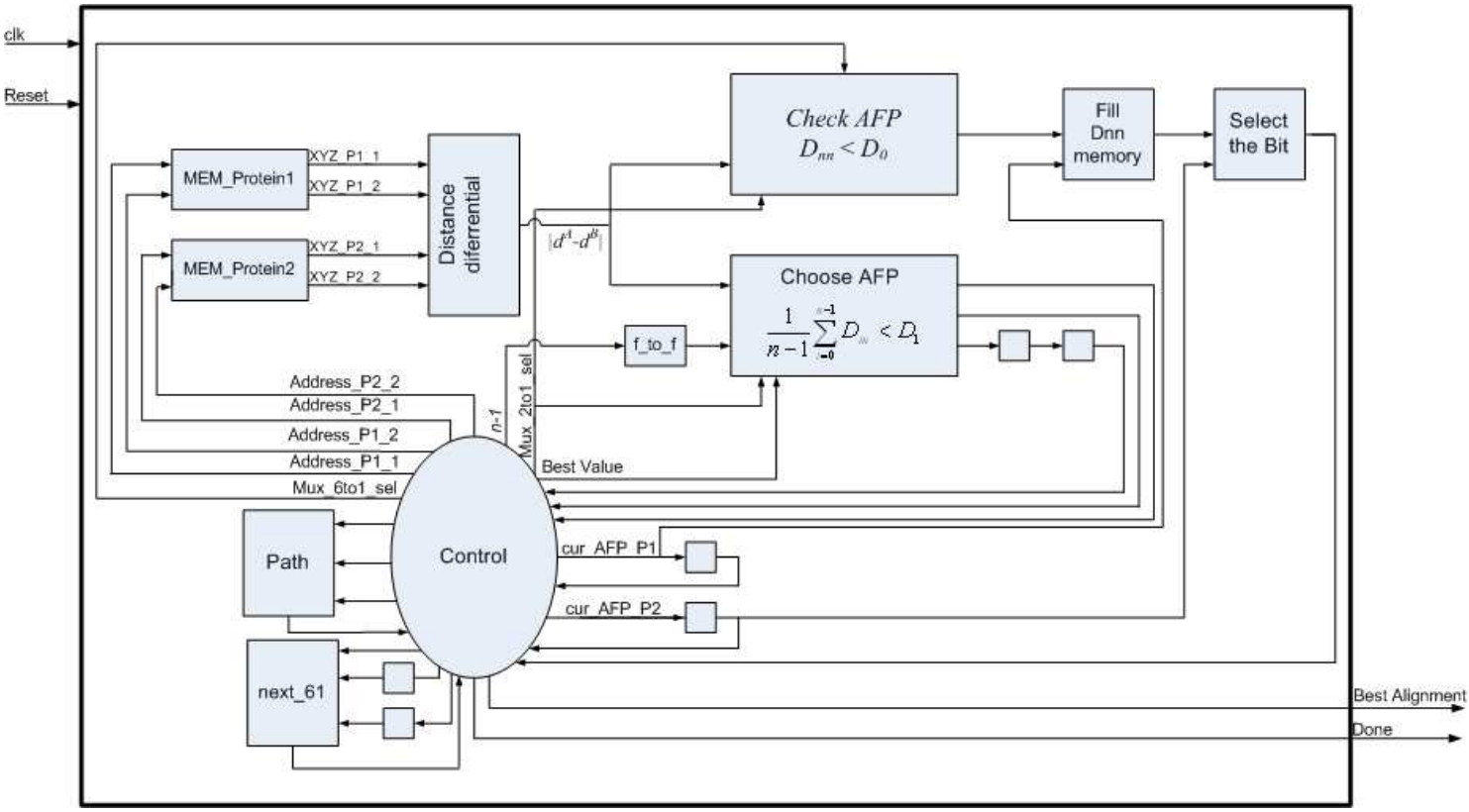
Το κεφάλαιο αυτό παρουσιάζει την διαδικασία σχεδίασης και υλοποίησης του τμήματος του αλγόριθμου CE που υλοποιήθηκε σε αναδιατασσόμενη λογική.

4.1 Εισαγωγή στην σχεδίαση και την υλοποίηση

Σχεδιάστηκε και υλοποιήθηκε μια αρχιτεκτονική που υλοποιεί ένα τμήμα του αλγορίθμου CE το οποίο ελέγχει όλα τα πιθανά AFPs που μπορεί να δημιουργηθούν από δύο πρωτεΐνες και δημιουργεί όλα τα μονοπάτια ομοιότητας των πρωτεϊνών έτσι ώστε να βρεθεί το μέγιστο μονοπάτι ομοιότητας.

Ο σχεδιασμός βασίστηκε στα βήματα που ακολουθήθηκαν κατά την μοντελοποίηση του αλγορίθμου. Τα βασικά μέρη που απαρτίζουν την αρχιτεκτονική είναι οι μονάδες που εκτελούν τις αριθμητικές πράξεις και τις συγκρίσεις τιμών, οι μνήμες και η μονάδα ελέγχου.

Η κύρια μονάδα τη αρχιτεκτονικής δέχεται ως είσοδο τα δεδομένα που είναι φορτωμένα σε δύο μνήμες, το σήμα ρολογιού και το σήμα Reset. Η έξοδος αποτελείται από ένα σήμα 32-bit που αντιπροσωπεύει το μεγαλύτερο μέγεθος μονοπατιού ομοιότητας, αλλά και από ένα σήμα που ενεργοποιείται όταν ολοκληρωθεί η διαδικασία.



Σχήμα 4.1: Σε αυτό το σχήμα παρατηρείται η αρχιτεκτονική της σχεδίασης που υπολογίζει το μεγαλύτερο μονοπάτι ομοιότητας της τριτοταγούς δομής δύο πρωτεϊνών.

4.2 Βασικές δομικές μονάδες της σχεδίασης και της υλοποίησης

4.2.1 Επιλογή της FPGA και του λογισμικού

Η αρχιτεκτονική που υλοποιήθηκε είναι ιδιαίτερα απαιτητική κυρίως σε πράξεις κινητής υποδιαστολής. Για τον λόγο αυτό επιλέχθηκε η FPGA VSX240T της οικογένειας Virtex 5, λόγω του ότι διαθέτει τον μεγαλύτερο αριθμό σε DSP48E slices, τα οποία εκτελούν πράξεις κινητής υποδιαστολής, σε σχέση με τις υπόλοιπες FPGA που διατίθενται σε Virtex 5. Η υλοποίηση έγινε σε γλώσσα VHDL και χρησιμοποιήθηκε το εργαλείο Xilinx Ise Design Suite 10.1. Για την προσομοίωση χρησιμοποιήθηκε το πρόγραμμα Modelsim SE 6.3f.

4.2.2 Αναπαράσταση αριθμών κινητής υποδιαστολής

Οι αριθμητικές πράξεις που εκτελούνται στην συγκεκριμένη αρχιτεκτονική είναι πράξεις κινητής υποδιαστολής απλής ακρίβειας και οι αριθμοί αναπαριστώνται σε δυαδική μορφή σύμφωνα με τον τρόπο αναπαράστασής τους από το πρότυπο IEEE-745. Η αναπαράσταση ενός αριθμού 32-bit σύμφωνα με αυτό το πρότυπο φαίνεται πιο κάτω:

```
 S    EEEEEEEE  FFFFFFFFFFFFFFFFFFFFFFF
31    30        23                     0
+/-   exponent  Fraction
```

Βασικά, ένας αριθμός κινητής υποδιαστολής αποτελείται από το bit πρόσημου (+/-), ένα κανονικοποιημένο εκθέτη και το κλασματικό μέρος. Για την μετατροπή μίας αναπαράστασης 32-bit σε έναν αριθμό χρησιμοποιείται η ακόλουθη εξίσωση:

S * 2 ^ (exponent – exponent_base) * (1.0 + Fraction / fraction_base)

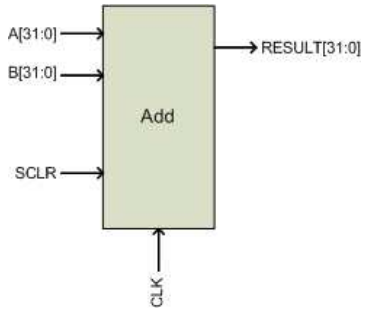
όπου exponent_base = 2^((maximum exponent/2)-1)

και fraction_base = (maximum possible unsigned fraction+1).

4.2.3 Μονάδες εκτέλεσης πράξεων κινητής υποδιαστολής απλής ακρίβειας

Για την εκτέλεση των αριθμητικών πράξεων σε αυτή τη σχεδίαση χρησιμοποιήθηκαν αθροιστές, αφαιρέτες, πολλαπλασιαστές, διαιρέτες, αλλά και μονάδες υπολογισμού τετραγωνικής ρίζας καθώς και συγκριτές κινητής υποδιαστολής απλής ακρίβειας. Επίσης χρησιμοποιήθηκε και μία μονάδα μετατροπής αριθμού σταθερής υποδιαστολής σε αριθμό κινητής υποδιαστολής απλής ακρίβειας. Οι μονάδες αυτές δημιουργήθηκαν με το εργαλείο Xilinx Core Generator 10.1. Πιο κάτω παρουσιάζονται οι διεπαφές των μονάδων όπως αυτές επιλέχθηκαν για την υλοποίηση της σχεδίασης, αλλά και οι επιλογές που έγιναν κατά την δημιουργία τους, η επιλογή των οποίων θα τεκμηριωθεί στην συνέχεια κατά την επεξήγηση της σχεδιαστικής διαδικασίας.

Αθροιστής



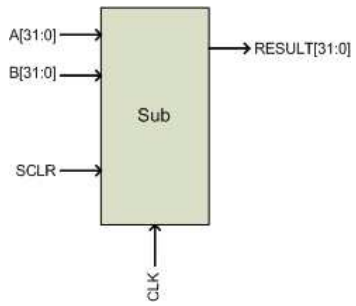
Σχήμα 4.2.3.1: Διεπαφή αθροιστή

Σήμα	*Τύπος*	*Πλάτος*	*Περιγραφή*
A	Είσοδος	32	Τελεστής A
B	Είσοδος	32	Τελεστής B
SCLR	Είσοδος	1	Συνχρονιζόμενο Reset
CLK	Είσοδος	1	Ρολόι
RESULT	Έξοδος	32	Αποτέλεσμα: Result = A + B

Πίνακας 4.2.3.1: Σήματα εισόδου και εξόδου αθροιστή

Στην σχεδίαση χρησιμοποιούνται 16 αθροιστές με latency 8 και ο κάθε αθροιστής χρησιμοποιεί 2 DSP48Es. Στις μονάδες υπολογισμού ευκλείδειας απόστασης χρησιμοποιούνται συνολικά 6 αθροιστές, για τον υπολογισμό του Dnn χρησιμοποιούνται 2 και για τον υπολογισμό του κριτηρίου επιλογής χρησιμοποιούνται 8.

Αφαιρέτης



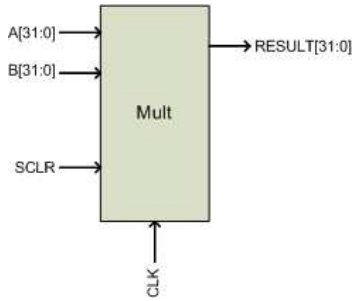
Σχήμα 4.2.3.2: Διεπαφή αφαιρέτη

Σήμα	*Τύπος*	*Πλάτος*	*Περιγραφή*
A	Είσοδος	32	Τελεστής A
B	Είσοδος	32	Τελεστής B
SCLR	Είσοδος	1	Συνχρονιζόμενο Reset
CLK	Είσοδος	1	Ρολόι
RESULT	Έξοδος	32	Αποτέλεσμα: Result = A - B

Πίνακας 4.2.3.2: Σήματα εισόδου και εξόδου αφαιρέτη

Στην σχεδίαση χρησιμοποιούνται 7 αφαιρέτες με latency 11 (maximum latency) και ο κάθε αφαιρέτης χρησιμοποιεί 2 DSP48Es. Στις δύο μονάδες υπολογισμού της ευκλείδειας απόστασης χρησιμοποιούνται 6 αφαιρέτες και για την διαφορά των ευκλείδειων αποστάσεων χρησιμοποιείται ακόμα 1 αφαιρέτης.

Πολλαπλασιαστής



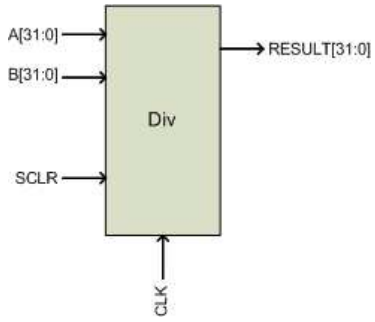
Σχήμα 4.2.3.3: Διεπαφή πολλαπλασιαστή

Σήμα	*Τύπος*	*Πλάτος*	*Περιγραφή*
A	Είσοδος	32	Τελεστής A
B	Είσοδος	32	Τελεστής B
SCLR	Είσοδος	1	Συνχρονιζόμενο Reset
CLK	Είσοδος	1	Ρολόι
RESULT	Έξοδος	32	Αποτέλεσμα: Result = A * B

Πίνακας 4.2.3.3: Σήματα εισόδου και εξόδου πολλαπλασιαστή

Στην σχεδίαση χρησιμοποιούνται 6 πολλαπλασιαστές, για την υλοποίηση των μονάδων υπολογισμού της ευκλείδειας απόστασης, με latency 6 (maximum latency) και ο κάθε πολλαπλασιαστής χρησιμοποιεί 3 DSP48Es.

Διαιρέτης



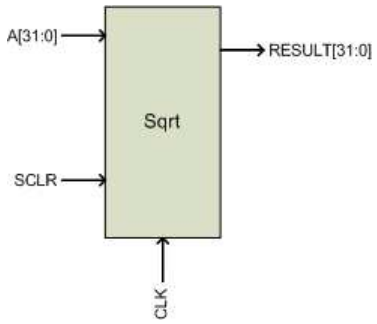
Σχήμα 4.2.3.4: Διεπαφή διαιρέτη

Σήμα	*Τύπος*	*Πλάτος*	*Περιγραφή*
A	Είσοδος	32	Τελεστής A
B	Είσοδος	32	Τελεστής B
SCLR	Είσοδος	1	Συνχρονιζόμενο Reset
CLK	Είσοδος	1	Ρολόι
RESULT	Έξοδος	32	Αποτέλεσμα: Result = A / B

Πίνακας 4.2.3.4: Σήματα εισόδου και εξόδου διαιρέτη

Στην σχεδίαση χρησιμοποιούνται 3 διαιρέτες με latency 28 (maximum latency). Ένας διαιρέτης χρησιμοποιείται για τον υπολογισμό του Dnn και 2 για τον υπολογισμό του κριτηρίου επιλογής. Στην περίπτωση του διαιρέτη δεν υπάρχει η δυνατότητα επιλογής χρήσης πόρων από τα DSP48E slices που διατίθενται.

Τετραγωνική ρίζα



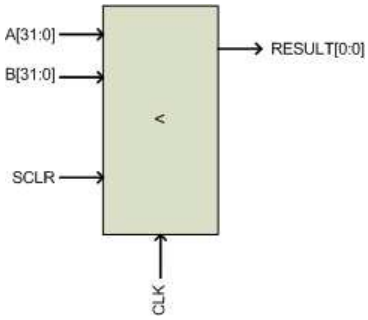
Σχήμα 4.2.3.5: Διεπαφή μονάδας υπολογισμού τετραγωνικής ρίζας

Σήμα	*Τύπος*	*Πλάτος*	*Περιγραφή*
A	Είσοδος	32	Τελεστής A
SCLR	Είσοδος	1	Συνχρονιζόμενο Reset
CLK	Είσοδος	1	Ρολόι
RESULT	Έξοδος	32	Αποτέλεσμα: Result = sqrt(A)

Πίνακας 4.2.3.5: Σήματα εισόδου και εξόδου της μονάδας υπολογισμού τετραγωνικής ρίζας

Στην σχεδίαση χρησιμοποιούνται 2 μονάδες υπολογισμού τετραγωνικής ρίζας με latency 28 (maximum latency) για τον υπολογισμό των ευκλείδειων αποστάσεων. Και στην περίπτωση της μονάδας υπολογισμού τετραγωνικής ρίζας, όπως και του διαιρέτη, δεν υπάρχει η δυνατότητα επιλογής χρήσης πόρων από τα DSP48E slices που διατίθενται.

Συγκριτής – Less than



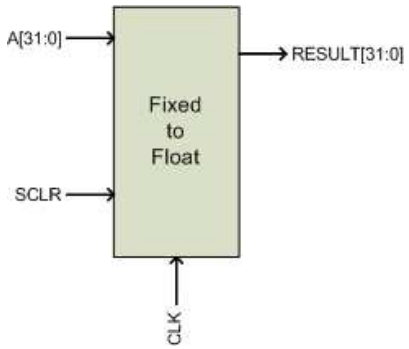
Σχήμα 4.2.3.6: Διεπαφή συγκριτή

Σήμα	*Τύπος*	*Πλάτος*	*Περιγραφή*
A	Είσοδος	32	Τελεστής A
B	Είσοδος	32	Τελεστής B
SCLR	Είσοδος	1	Συνχρονιζόμενο Reset
CLK	Είσοδος	1	Ρολόι
RESULT	Έξοδος	1	Αποτέλεσμα: Result = 1 αν A < B

Πίνακας 4.2.3.6: Σήματα εισόδου και εξόδου συγκριτή

Στην σχεδίαση χρησιμοποιούνται 3 συγκριτές, 1 για την σύγκριση του Dnn με το D_0 και 2 για την σύγκριση του κριτηρίου επιλογής με το D_1 και την μέχρι στιγμής μικρότερη τιμή του, με latency 2 (maximum latency). Κατά την δημιουργία του επιλέχθηκε η ιδιότητα “less than” σύμφωνα με την οποία η έξοδος του είναι 1 αν η είσοδος A είναι μικρότερη από την B. Και σε αυτή την περίπτωση δεν παρέχεται η δυνατότητα χρήσης των DSP48Es.

Μετατροπέας αριθμού σταθερής υποδιαστολής σε αριθμό κινητής υποδιαστολής (fixed to float)



Σχήμα 4.2.3.7: Διεπαφή μονάδας fixed to float.

Σήμα	*Τύπος*	*Πλάτος*	*Περιγραφή*
A	Είσοδος	32	Είσοδος A σε μορφή fixed
SCLR	Είσοδος	1	Συνχρονιζόμενο Reset
CLK	Είσοδος	1	Ρολόι
RESULT	Έξοδος	32	Αποτέλεσμα: Result = float(A)

Πίνακας 4.2.3.7: Σήματα εισόδου και εξόδου της μονάδας fixed to float

Στην σχεδίαση χρησιμοποιείται μία μονάδα fixed to float (f_to_f) με latency 6 (maximum latency). Η είσοδος A που αντιπροσωπεύει έναν αριθμό σταθερής υποδιαστολής είναι ένας ακέραιος αριθμός και για τον λόγο αυτό επιλέχτηκε η είσοδος A να αποτελείται μόνο από το ακέραιο μέρος μήκους 32 bits, ενώ ως έξοδος Result επιλέχθηκε ένας αριθμός κινητής υποδιαστολής απλής ακρίβειας μήκους 32 bits επίσης. Η μονάδα αυτή δημιουργήθηκε μόνο με την χρήση λογικής, κάτι που αποτελούσε και την μόνη επιλογή για την χρήση των πόρων κατά την δημιουργία της.

Σε όλες τις μονάδες έχει επιλεγεί το μέγιστο latency με στόχο η συχνότητα λειτουργίας του συστήματος να είναι όσο το δυνατόν υψηλότερη. Η μόνη περίπτωση στην οποία επιλέχθηκε μία άλλη τιμή είναι ο αθροιστής, όπου ενώ το μέγιστο latency που μπορεί να επιλεγεί είναι 11, επιλέχθηκε ως ιδανικότερο το 8 το οποίο παρέχει μία υψηλή συχνότητα και εξυπηρετεί στην λύση σχεδιαστικών προβλημάτων. Όλες οι μονάδες έχουν την δυνατότητα να λειτουργούν ομόχειρα δίνοντας ένα αποτέλεσμα σε

κάθε κύκλο. Αθροίζοντας τον αριθμό των DSP48Es χρησιμοποιήθηκαν παρατηρείται ότι μπορεί να εξυπηρετηθεί και από άλλες FPGAs, όμως ο μεγάλος αριθμός που διαθέτει η VSX240T που επιλέχθηκε επιτρέπει βελτιστοποιήσεις της απόδοσης με την χρήση παράλληλων μονάδων.

4.3 Μνήμη BRAM

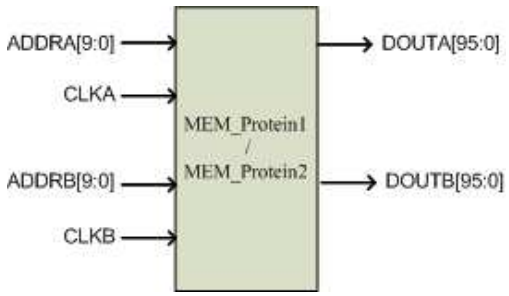
Όσον αφορά τα διαθέσιμα block RAM blocks η VSX240T διαθέτει 516 blocks των 36Kb που μπορούννα χρησιμοποιηθούν ως 1032 ανεξάρτητα blocks των 18Kb. Και στις δύο περιπτώσεις τα διαθέσιμα block RAM blocks ικανοποιούν τις απαιτήσεις της υλοποίησης. Για την υλοποίηση, από αυτά τα block ram blocks δημιουργήθηκαν πέντε μνήμες με την χρήση του Xilinx Core Generator 10.1.

4.3.1 Μνήμες MEM_Protein1 και MEM_Protein2

Οι μνήμες με τις ονομασίες MEM_Protein1 και MEM_Protein2 είναι οι μνήμες που περιέχουν τις τιμές για τις συντεταγμένες X, Y, Z, των C_α ατόμων για το κάθε κατάλοιπο και αποτελούν τις εισόδους του συστήματος. Η MEM_Protein1 έχει τα δεδομένα που αφορούν την πρώτη πρωτεΐνη και η MEM_Protein2 τα δεδομένα που αφορούν την δεύτερη πρωτεΐνη. Οι μνήμες αυτές είναι dual port ROM και έχουν βάθος 1024 θέσεων των 96 bits ανά θέση μνήμης. Ο αριθμός των 1024 θέσεων είναι μια γενική τιμή που επιλέχθηκε έτσι ώστε να καλύπτονται όλες οι περιπτώσεις ανάλογα με το μέγεθος της εισόδου. Κάθε θέση της κάθε μνήμης έχει πλάτος 96 bits τα οποία χωρισμένα σε ομάδες των 32 bits αντιστοιχούν στα X, Y, Z. Κάθε ομάδα των 32 bits, αντιπροσωπεύει έναν αριθμό κινητής υποδιαστολής που αναπαρίσταται σε δυαδική μορφή σύμφωνα με το πρότυπο IEEE-745.

Σχήμα 4.3.1.1: Σε αυτό το σχήμα φαίνεται ποία από τα 96 bits μια θέσης μνήμης, MEM_Protein1 και MEM_Protein2, στα X, Y, και Z μεγέθους 32 bits το καθένα.

Η πηγή από την οποία πάρθηκαν τα δεδομένα που περιέχονται στις μνήμες αυτές είναι *PDB* αρχεία από την Protein Data Bank. Τα αρχεία αυτά διαβάζονται με μία εφαρμογή που δημιουργήθηκε κατά το στάδιο της μοντελοποίησης σε γλώσσα προγραμματισμού C και κατά την διαδικασία αυτή επιλέγονται από τα *PDB* αρχεία οι πληροφορίες που είναι χρήσιμες, μετατρέπονται στην επιθυμητή μορφή σύμφωνα με τον τρόπο της δυαδικής αναπαράστασης που επιλέχθηκε και αποθηκεύονται σε αρχεία με τον κατάλληλο τρόπο έτσι ώστε να δημιουργούνται τα δύο coe αρχεία που θα χρησιμοποιηθούν για την αρχικοποίηση αυτών των μνημών.



Σχήμα 4.3.1.2: Διεπαφή των μνημών MEM_Protein1 και MEM_Protein2.

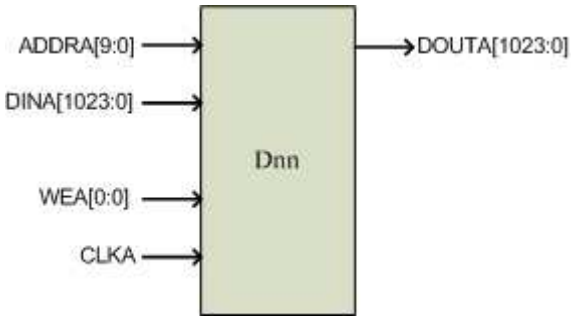
Σήμα	*Τύπος*	*Πλάτος*	*Περιγραφή*
ADDRA	Είσοδος	10	Διεύθυνση θύρας A
CLKA	Είσοδος	1	Ρολόι θύρας A
ADDRB	Είσοδος	10	Διεύθυνση θύρας B
CLKB	Είσοδος	1	Ρολόι θύρας B
DOUTA	Έξοδος	96	Έξοδος θύρας A
DOUTB	Έξοδος	96	Έξοδος θύρας B

Πίνακας 4.3.1.1: Σήματα εισόδου και εξόδου των μνημών MEM_Protein1 και MEM_Protein2.

4.3.2 Μνήμη Dnn

Η μνήμη με την ονομασία Dnn είναι μια μνήμη στην οποία κάθε bit αντιστοιχεί σε ένα AFP. Εάν η τιμή του bit είναι 1 τότε αυτό το AFP μπορεί να χρησιμοποιηθεί ως μέρος ενός μονοπατιού, ενώ αν είναι 0 δεν μπορεί. Η τιμή αυτή

επιλέγεται σύμφωνα με το αποτέλεσμα της σχέσης 3.2.1. Η μνήμη αυτή είναι single port RAM και έχει βάθος 1024 θέσεων των 1024 bits ανά θέση μνήμης. Και πάλι οι τιμές 1024 επιλέχθηκαν έτσι ώστε να μπορούν να ικανοποιήσουν όλα τα μεγέθη των εισόδων και γεμίζονται μόνο όσες θέσεις χρειάζονται ανάλογα με την περίπτωση. Οι θέσεις μνήμης αντιστοιχούν στην αρχική θέση, p^A, του AFP στην πρωτεΐνη A και ο αριθμός που αντιστοιχεί στο κάθε bit της κάθε θέσης της μνήμης αντιστοιχεί στην θέση, p^B, του AFP στην πρωτεΐνη B.



Σχήμα 4.3.2.1: Διεπαφή της μνήμης Dnn.

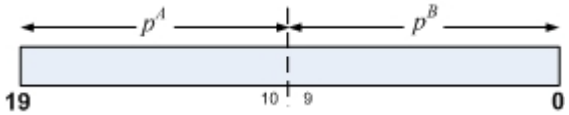
Σήμα	*Τύπος*	*Πλάτος*	*Περιγραφή*
ADDRA	Είσοδος	10	Διεύθυνση θέσης στην μνήμη
DINA	Είσοδος	1024	Δεδομένα προς εγγραφή στην μνήμη
WEA	Είσοδος	1	Σήμα ενεργοποίησης εγγραφής στην μνήμη
CLKA	Είσοδος	1	Ρολόι
DOUTA	Έξοδος	1024	Έξοδος μνήμης

Πίνακας 4.3.2.1: Σήματα εισόδου και εξόδου της μνήμης Dnn.

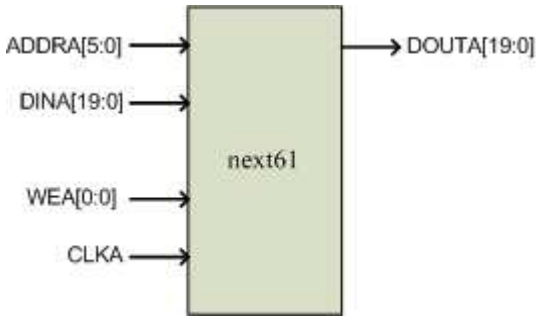
4.3.3 Μνήμη next61

Η μνήμη με την ονομασία next61 είναι μια μνήμη τύπου single port RAM βάθους 61 θέσεων των 20 bits ανά θέση μνήμης. Κατά την διαδικασία στην οποία προστίθεται ένα AFP σε ένα μονοπάτι, ελέγχονται και τα 61 από τα υποψήφια AFPs αν μπορούν να αποτελέσουν μέρος ενός μονοπατιού σύμφωνα με το κριτήριο που

ελέγχει η σχέση 3.2.1. Όσα από αυτά τα AFPs πληρούν το κριτήριο αυτό αποθηκεύονται στην μνήμη next61. Έτσι εξηγείται γιατί επιλέχθηκε το βάθος την μνήμης να είναι 61, αφού ο μέγιστος αριθμός των AFPs που μπορούν να είναι υποψήφια και να πληρούν το κριτήριο ύπαρξης τους ως AFPs ενός μονοπατιού είναι 61. Σε μία θέση μνήμης αποθηκεύεται η διεύθυνση της αρχικής θέσης του AFP στην πρωτεΐνη A (p^A) και η διεύθυνση της αρχικής θέσης του AFP στην πρωτεΐνη B (p^B). Λόγω της παραδοχής που έγινε ότι το μέγιστο μέγεθος των πρωτεϊνών που θα αποτελούν τις εισόδους του συστήματος είναι 1024 η κάθε διεύθυνση έχει μέγεθος 10 bits και γι' αυτό το λόγο το πλάτος της μνήμης είναι 20 bits.



Σχήμα 4.3.3.1: Σε αυτό το σχήμα φαίνεται ποια από τα 20 bits μια θέσης της μνήμης, next61, αντιστοιχεί στα p^A και p^B.



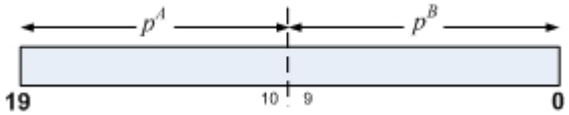
Σχήμα 4.3.3.2: Διεπαφή της μνήμης next61.

Σήμα	*Τύπος*	*Πλάτος*	*Περιγραφή*
ADDRA	Είσοδος	6	Διεύθυνση θέσης στην μνήμη
DINA	Είσοδος	20	Δεδομένα προς εγγραφή στην μνήμη
WEA	Είσοδος	1	Σήμα ενεργοποίησης εγγραφής στην μνήμη
CLKA	Είσοδος	1	Ρολόι
DOUTA	Έξοδος	20	Έξοδος μνήμης

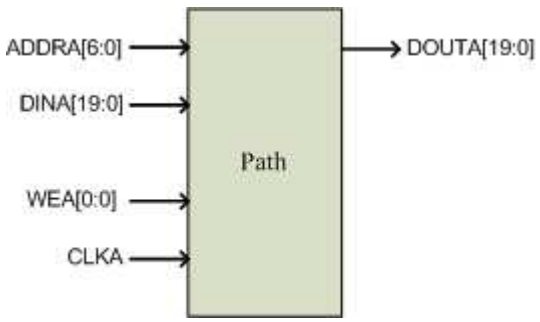
Πίνακας 4.3.3.1: Σήματα εισόδου και εξόδου της μνήμης next61.

4.3.4 Μνήμη Path

Κατά την δημιουργία ενός μονοπατιού αποθηκεύονται στην μνήμη Path οι αρχικές θέσεις στην πρωτεΐνη A και B, p^A και p^B αντίστοιχα, για το κάθε AFP που έχει ήδη προστεθεί στο μονοπάτι. Η μνήμη Path είναι τύπου single port RAM βάθους 128 θέσεων των 20 bits ανά θέση μνήμης, αφού όπως και στην next61 η κάθε διεύθυνση, p^A και p^B, έχει μέγεθος 10 bits και γι' αυτό το λόγο το πλάτος της μνήμης είναι 20 bits. Το βάθος επιλέχθηκε να είναι 128 διότι με βάση την υπόθεση ότι το μέγιστο μέγεθος των πρωτεϊνών είναι 1024, τότε σε ένα μονοπάτι μπορούν να υπάρχουν μέχρι και 128 AFPs (1024/8).



Σχήμα 4.3.4.1: Σε αυτό το σχήμα φαίνεται ποια από τα 20 bits μια θέσης της μνήμης, Path, αντιστοιχεί στα p^A και p^B.



Σχήμα 4.3.4.2: Διεπαφή της μνήμης Path.

Σήμα	*Τύπος*	*Πλάτος*	*Περιγραφή*
ADDRA	Είσοδος	7	Διεύθυνση θέσης στην μνήμη
DINA	Είσοδος	20	Δεδομένα προς εγγραφή στην μνήμη
WEA	Είσοδος	1	Σήμα ενεργοποίησης εγγραφής στην μνήμη
CLKA	Είσοδος	1	Ρολόι
DOUTA	Έξοδος	20	Έξοδος μνήμης

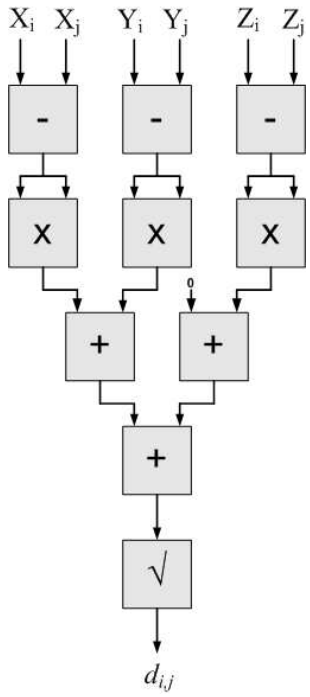
Πίνακας 4.3.4.1: Σήματα εισόδου και εξόδου της μνήμης Path.

4.4 Μονάδα υπολογισμού ευκλείδειας απόστασης

Ο υπολογισμός της απόστασης *d* μεταξύ δύο κατάλοιπων μιας πρωτεΐνης είναι απαραίτητο στοιχείο για την υλοποίηση όλων των τύπων που περιλαμβάνει ο αλγόριθμος CE, όπως είναι οι 3.2.1 και 3.2.7 που χρειάζονται για το μέρος του αλγορίθμου που υλοποιήθηκε και αποτελεί την ευκλείδεια απόσταση μεταξύ των C_a ατόμων των δύο κατάλοιπων. Η ευκλείδεια απόσταση $d_{i,j}$ μεταξύ των καταλοίπων *i* και j υπολογίζεται ως εξής.

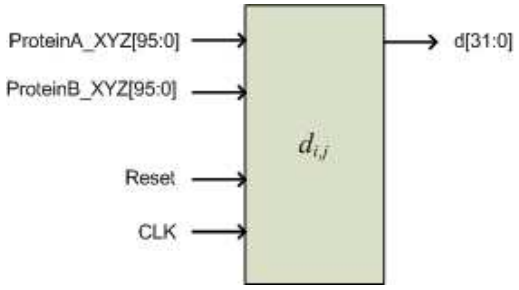
$$d_{i,j} = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2 + (Z_i - Z_j)^2} \qquad (4.3.1)$$

Για την υλοποίηση αυτής της σχέσης χρειάζονται 3 αφαιρέτες, 3 πολλαπλασιαστές, 3 αθροιστές και μία μονάδα υπολογισμού τετραγωνικής ρίζας. Όλες αυτές οι μονάδες έχουν δημιουργηθεί σύμφωνα με τις προδιαγραφές που περιγράφονται πιο πριν και σχηματίζουν την ακόλουθη διάταξη.



Σχήμα 4.4.1: Η διάταξη των μονάδων που απαρτίζουν μία μονάδα υπολογισμού $d_{i,j.}$

Όπως φαίνεται και στο πιο πάνω σχήμα για τον υπολογισμό του τετραγώνου της διαφοράς γίνεται ένας πολλαπλασιασμός της διαφοράς με τον εαυτό της και γι' αυτό χρησιμοποιείται ένας πολλαπλασιαστής με τις δύο εισόδους του να είναι το αποτέλεσμα της διαφοράς. Επειδή υπάρχουν τρία αποτελέσματα από τους πολλαπλασιαστές τα οποία πρέπει να προστεθούν μεταξύ τους και ο αθροιστής έχει δύο εισόδους χρησιμοποιείται η διάταξη με τους τρεις αθροιστές που φαίνεται στο σχήμα 4.4.1. Ο ένας αθροιστής που προσθέτει το αποτέλεσμα του ενός πολλαπλασιαστή με το 0 χρησιμεύει για να καθυστερήσει το αποτέλεσμα τόσους κύκλους όσους χρειάζεται για να δοθεί το αποτέλεσμα της πρόθεσης των εξόδων των άλλων πολλαπλασιαστών, έτσι ώστε να φτάσουν ταυτόχρονα στον τελευταίο αθροιστή που θα δώσει το συνολικό άθροισμα. Εναλλακτικά στην θέση του αθροιστή, με την μία είσοδο στο 0, μπορούν να χρησιμοποιηθούν καταχωρητές που θα επιφέρουν την επιθυμητή καθυστέρηση στην έξοδο του πολλαπλασιαστή.



Σχήμα 4.4.2: Διεπαφή μονάδας υπολογισμού $d_{i,j.}$

Σήμα	*Τύπος*	*Πλάτος*	*Περιγραφή*
ProteinA_XYZ	Είσοδος	96	Έξοδος από την μνήμη MEM_Protein1. Αντιστοιχεί στις συντεταγμένες X, Y, Z για την πρωτεΐνη A.
ProteinB_XYZ	Είσοδος	96	Έξοδος από την μνήμη MEM_Protein2. Αντιστοιχεί στις συντεταγμένες X, Y, Z για την πρωτεΐνη B.
Reset	Είσοδος	1	Reset προς τις floating-point μονάδες
CLK	Είσοδος	1	Ρολόι
d	Έξοδος	32	Το αποτέλεσμα $d_{i,j}$

Πίνακας 4.4.1: Σήματα εισόδου και εξόδου της μονάδας υπολογισμού $d_{i,j.}$

4.5 Μονάδα υπολογισμού του D_{nn} και σύγκρισής του με το D_o

Για την υλοποίηση της σχέσης 3.2.1, $Dnn < D_0$, χρειάζεται αρχικά να σχεδιαστεί μια διάταξη που να υπολογίζει το D_{nn} χρησιμοποιώντας τον τύπο (3.3.1) ο οποίος επαναλαμβάνεται πιο κάτω:

$$Dij = \frac{1}{21}\left(\sum_{k=0}^{m-2}\sum_{l=k+2}^{m}\left|d^{A}_{p_i^A+k,\,p_j^A+l} - d^{B}_{p_i^B+k,\,p_j^B+l}\right|\right)$$

Παρατηρώντας τον τύπο είναι φανερό ότι για την υλοποίησή του χρειάζονται δύο μονάδες υπολογισμού του $d_{i,j}$, και ένας αφαιρέτης που να υπολογίζει την διαφορά τους. Όσον αφορά την απόλυτη τιμή αυτής της διαφοράς υλοποιείται ως μία λογική που παρεμβάλλεται μεταξύ των αποτελεσμάτων από τις δύο μονάδες $d_{i,j}$ και του

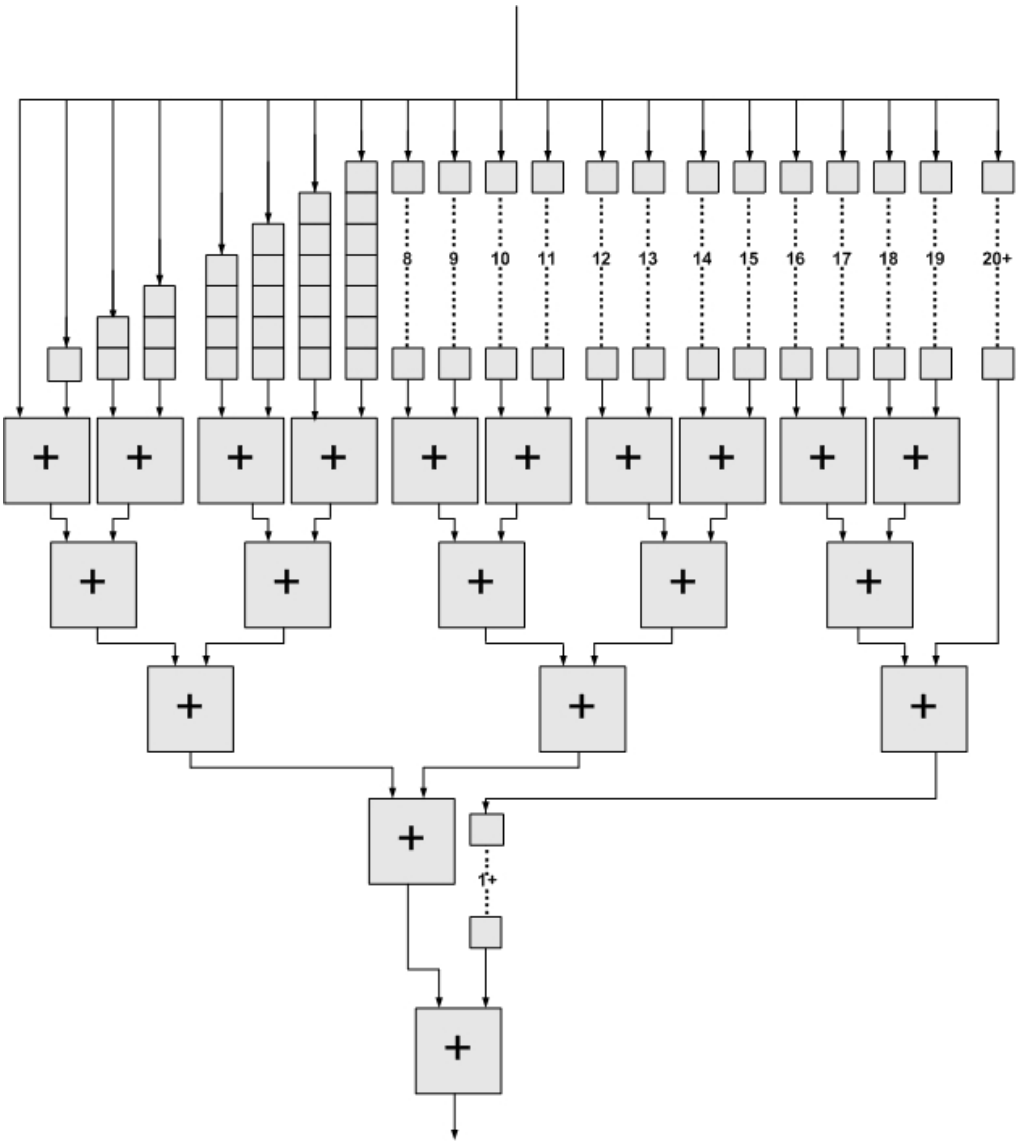
αφαιρέτη. Η λογική αυτή συγκρίνει τις εξόδους των μονάδων $d_{i,j}$ και οδηγεί την μεγαλύτερη στην είσοδο A του αφαιρέτη και την μικρότερη στην είσοδο B. Χρησιμοποιείται ακόμη ένας διαιρέτης που εκτελεί την διαίρεση με το 21 το οποίο είναι σταθερή είσοδος B στον διαιρέτη και αναπαρίσταται στην μορφή αριθμού κινητής υποδιαστολής 32-bit σύμφωνα με το πρότυπο IEEE-745.

Το πιο απαιτητικό μέρος στην σχεδίαση όσο αφορά τον τύπο 3.3.1 είναι το διπλό άθροισμα. Έχοντας αποφασιστεί όσον αφορά αυτή την εργασία ότι το *m* ;έχει την τιμή 8, τα όρια των αθροισμάτων υποδεικνύουν ότι πρέπει να αθροιστούν 21 διαφορές ($|d^A-d^B|$). Αναλυτικότερα για k=0 αντιστοιχούν 6 τιμές προς άθροιση, για k=1 αντιστοιχούν 5, για k=2 αντιστοιχούν 4, για k=3 αντιστοιχούν 3, k=4 αντιστοιχούν 2 και k=5 αντιστοιχεί 1 τιμή. Οπόταν για τον υπολογισμό μία τιμής D_{nn} πρέπει να προστεθούν μεταξύ τους 21 τιμές.

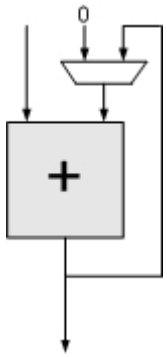
Με μία πλήρη εκμετάλλευση της δυνατότητας των μονάδων που χρησιμοποιήθηκαν για pipelining είναι δυνατόν κάθε κύκλο να υπολογίζεται και μία νέα τιμή D_{nn}. Μέχρι στιγμής αυτό είναι εφικτό αφού οι μνήμες MEM_Protein1 και MEM_Protein2 μπορούν εφόσον δέχονται διαφορετική είσοδο σε κάθε κύκλο να δίνουν και διαφορετική έξοδο σε κάθε κύκλο. Επίσης όλες οι πράξεις που γίνονται στην συνέχεια και αφορούν τον υπολογισμό των $d_{i,j}$ και της διαφοράς τους, μπορούν και αυτές κάθε κύκλο να δίνουν άλλο αποτέλεσμα. Έτσι κάθε κύκλο φτάνει στην μονάδα άθροισης μία διαφορά που αποτελεί μία από τις 21, θα προστεθεί με αυτές που προηγήθηκαν και το σύνολο τους στη συνέχεια θα προστεθεί με αυτή που θα φτάσει στον επόμενο κύκλο. Με αυτό τον τρόπο συνεχίζεται η διαδικασία για τον υπολογισμό του αθροίσματος. Όταν φτάσει και η εικοστή πρώτη τότε θα έρθει η πρώτη που αφορά μία άλλη τιμή D_{nn} και η διαδικασία θα συνεχιστεί με αυτό τον τρόπο.

Υπάρχουν δύο λύσεις με τις οποίες μπορεί να υλοποιηθεί το πρόβλημα του αθροίσματος που παρουσιάστηκε στην προηγούμενη παράγραφο. Η μία λύση είναι να χρησιμοποιηθεί η διάταξη που φαίνεται στο σχήμα 4.5.1, χρησιμοποιώντας 20 αθροιστές και περισσότερους από 210 καταχωρητές (ο ακριβής αριθμός των καταχωρητών εξαρτάται από το latency των αθροιστών) οι οποίοι ρυθμίζουν την ταυτόχρονη πρόσθεση των δεδομένων. Η ιδέα αυτή αν και είναι πολύ απλή καταναλώνει υπερβολικά μεγάλο ποσοστό πόρων και για το λόγο αυτό απορρίφθηκε αμέσως και δεν υλοποιήθηκε ποτέ. Η δεύτερη λύση είναι αυτή που παρουσιάζεται στο σχήμα 4.5.2, όπου χρησιμοποιείται ένας αθροιστής με ανατροφοδότηση, δηλαδή

η έξοδός του επιστρέφει στην μία από τις εισόδους του έτσι ώστε η καινούρια τιμή που φτάνει στην άλλη του είσοδο να προστίθεται με το άθροισμα των προηγούμενων. Σε αυτή την περίπτωση επιβάλλεται και η χρήση ενός πολυπλέκτη έτσι ώστε κάθε φορά που θα έρχεται το πρώτο από τα 21 στοιχεία που θα προστεθούν μεταξύ τους να προστίθεται με το 0επιλέγοντας το μέσω του πολυπλέκτη που βρίσκεται στην είσοδο με την ανατροφοδότηση. Στις υπόλοιπες περιπτώσεις θα επιλέγεται ως είσοδος που προέρχεται από την έξοδο του αθροιστή.



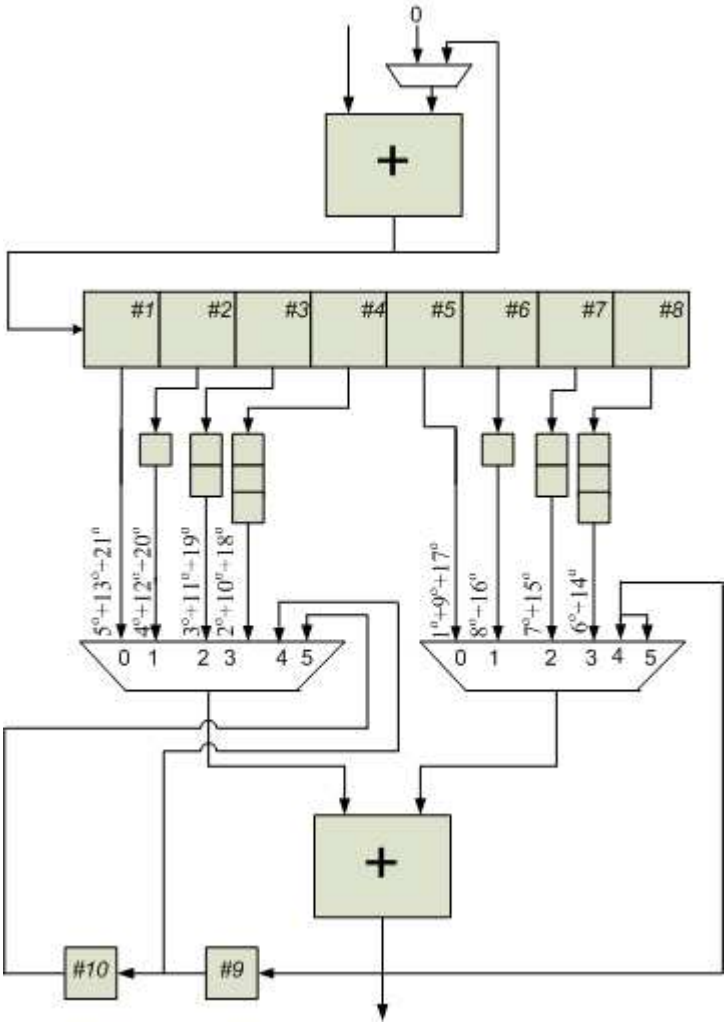
Σχήμα 4.5.1: Μια απλή σχεδιαστικά ιδέα για την υλοποίηση του διπλού αθροίσματος αλλά **<u>απορρίπτεται</u>** αμέσως λόγω της σπατάλης πολλών πόρων.



Σχήμα 4.5.2: Υλοποίηση του διπλού αθροίσματος χρησιμοποιώντας ανατροφοδότηση του αθροιστή.

Η λύση που παρουσιάζεται στο σχήμα 4.5.2, αν και χρειάζεται τους ελάχιστους πόρους που μπορούν να χρησιμοποιηθούν για την υλοποίηση του αθροίσματος, αναγκάζει το σύστημα να έχει μία χαμηλή συχνότητα λειτουργίας. Αυτό οφείλεται στο ότι ο μόνος τρόπος για να είναι λειτουργήσει η λύση του σχήματος 4.5.2, έτσι ώστε να υλοποιείται το διπλό άθροισμά χωρίς να υπάρχουν καθυστερήσεις σε κύκλους ρολογιού στο σύστημα είναι το latency του αθροιστή να είναι 1. Αυτό σε συνδυασμό με την ανατροφοδοτούμενη είσοδό του και τον πολυπλέκτη που παρεμβάλλεται μεταξύ αυτής και της εξόδου επιτρέπουν μία μέγιστη συχνότητα λειτουργίας του συστήματος περίπου στα 60MHz.

Μετά από δοκιμές παρατηρήθηκε ότι χρησιμοποιώντας την διάταξη του σχήματος 4.5.2 δίνεται μία ικανοποιητική συχνότητα λειτουργίας, περίπου 330MHz, όταν το latency του αθροιστή είναι ίσο με 8. Με βάση αυτή την παρατήρηση αποφασίστηκε ότι το latency του αθροιστή που θα χρησιμοποιείται σε όλη την σχεδίαση θα είναι ίσο με 8. Αυτή η απόφαση όμως οδηγεί σε κάποιες αλλαγές που πρέπει να γίνουν στην διάταξη του σχήματος 4.5.2 έτσι ώστε να παρέχεται η σωστή λειτουργία με την δυνατότητα για pipelining. Για τον λόγω αυτό έγινε μία αναπροσαρμογή της διάταξης του σχήματος 4.5.2 σε αυτή που αναπαριστάται στο σχήμα 4.5.3.



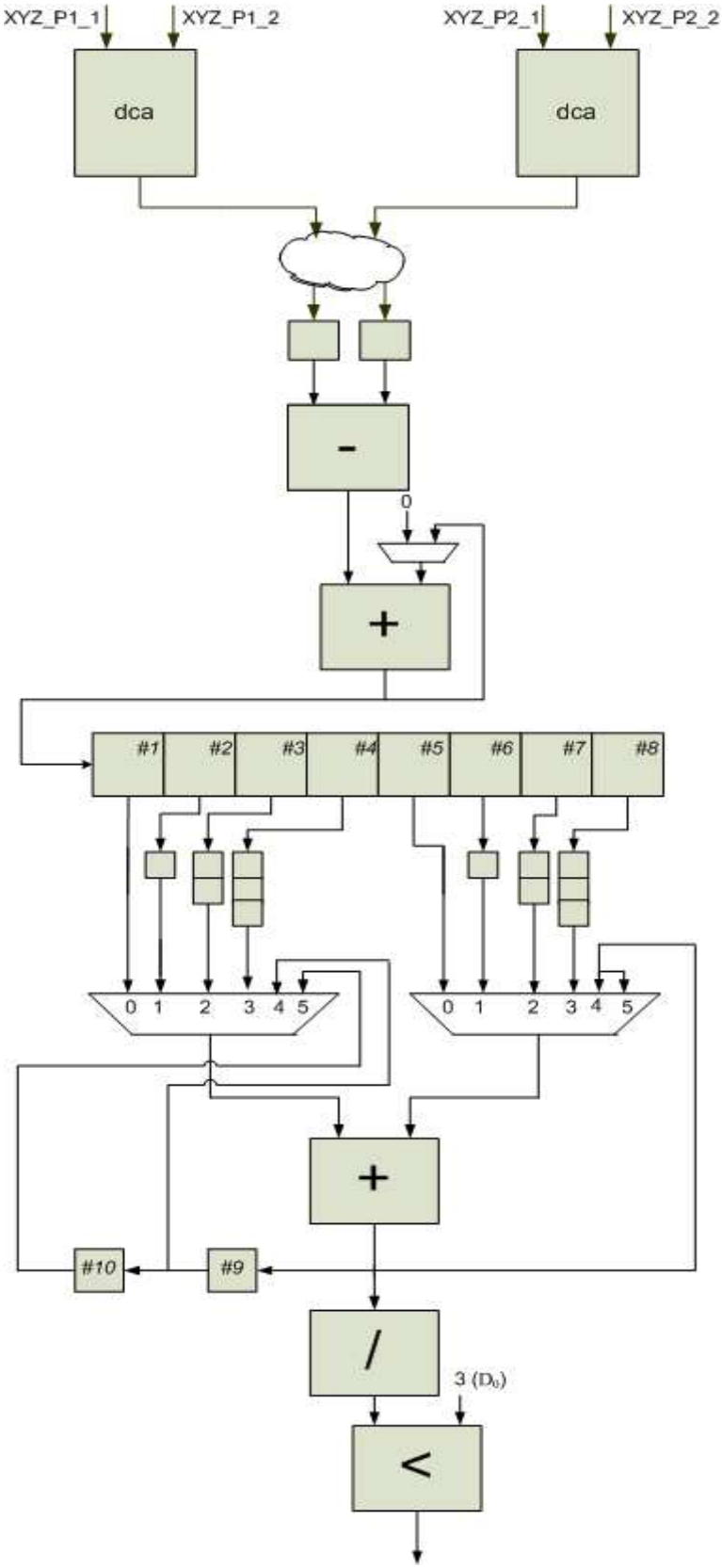
Σχήμα 4.5.3: Υλοποίηση του διπλού αθροίσματος χρησιμοποιώντας ανατροφοδότηση του αθροιστή με latency 8. Στις εξόδους των καταχωρητών φαίνονται ποια από τα 21 στοιχεία του αθροίσματος έχουν ήδη προστεθεί μεταξύ τους μέσω του αθροιστή με ανατροφοδότηση.

Στην λύση του προβλήματος που παρουσιάζεται στο σχήμα 4.5.3 δεν προστίθεται μόνο το 1ο από τα 21 στοιχεία με το 0. Λόγω του ότι το αποτέλεσμα της πρόσθεσης του 1^{ου} στοιχείου με το 0 δεν θα δοθεί από τον αθροιστή μετά από ένα κύκλο ώστε το αποτέλεσμα αυτό να προστεθεί με το 2^{ο}, το 2^{ο} αλλά και όλα τα υπόλοιπα στοιχεία που ακολουθούν μέχρι και το 8^{ο} προστίθενται και αυτά με το 0 το οποίο επιλέγεται ως είσοδος B του αθροιστή μέσω του πολυπλτέκτη που παρεμβάλλεται πριν την είσοδο B. Όταν δοθεί το αποτέλεσμα της πρόσθεσης του 1^{ου} στοιχείου από τον αθροιστή τότε αλλάζει πλέον το σήμα ελέγχου στου πολυπλέκτη και επιλέγεται ως σήμα εισόδου B του αθροιστή η έξοδός του. Με τον τρόπο αυτό αφού φτάσουν και τα 21 στοιχεία για να προστεθούν στον αθροιστή θα έχουν

προστεθεί και τα 21 σε οχτώ ομάδες όπως αυτές που φαίνονται στο σχήμα 4.5.3 στις εξόδους των καταχωρητών. Στην συνέχεια οι έξοδοι των καταχωρητών θα πρέπει να προστεθούν έγκαιρα μεταξύ τους πριν αλλοιωθούν από τα 21 επόμενα στοιχεία. Για να επιτευχθεί αυτό χρησιμοποιούνται και κάποιοι καταχωρητές επιπλέον όπως φαίνεται στο σχήμα.

Η πρόσθεση αυτή γίνεται μέσο του δεύτερου αθροιστή και ελέγχεται μέσο των δύο πολυπλεκτών 6 προς1 και των καταχωρητών με το νούμερο 9 και 10. Ο ένας από τους δύο πολυπλέκτες αν και θα μπορούσε να είναι 5 προς 1 αλλά λόγω του ότι επιλέχθηκε αυτοί οι δύο καταχωρητές να ελέγχονται από το ίδιο σήμα κατασκευάστηκε και αυτός ο πολυπλέκτης ως 6 προς 1 έχοντας την πέμπτη και έκτη είσοδό του ενωμένες στο ίδιο σήμα.

Έτσι στην αρχή προστίθενται η πρώτη είσοδος από τον πρώτο πολυπλέκτη με την πρώτη του δεύτερου ((5° + 13° + 12°) + (1° + 9° + 17°)). Στον επόμενο κύκλο περνάνε ως είσοδοι του αθροιστή οι δεύτερες είσοδοι και των δύο πολυπλεκτών ((4° + 12° + 20°) + (8° + 16°)). Στην συνέχεια δίνονται ως είσοδοι στον αθροιστή η τρίτη είσοδος τόσο από τον πρώτο, όσο και από τον δεύτερο πολυπλέκτη ((3° + 11° + 19°) + (7° + 15°)), ενώ στον επόμενο κύκλο σειρά έχουν η τέταρτη είσοδος του πρώτου και η τέταρτη είσοδος του δεύτερου αθροιστή ((2°+ 10° + 18°) + (6° + 14°)). Αφού περάσουν 8 κύκλοι από την στιγμή που έγινε η επιλογή των δεύτερων εισόδων των πολυπλεκτών ως είσοδοι του αθροιστή επιλέγονται οι πέμπτες είσοδοι των πολυπλεκτών για να προστεθούν μεταξύ τους(((5° + 13° + 12°) + (1° + 9° + 17°)) + ((4° + 12° + 20°) + (8° + 16°))) και μετά από δύο κύκλους επιλέγονται και πάλι οι ίδιες είσοδοι στους πολυπλέκτες(((3° + 11° + 19°) + (7° + 15°)) + 2°+ 10° + 18°) + (6° + 14°))).Και στις δύο περιπτώσεις ο συγχρονισμός των εισόδων στον αθροιστή λόγω του ότι η μία εξέρχεται από τον αθροιστή ένα κύκλο γρηγορότερα από την άλλη γίνεται μέσω του καταχωρητή με το νούμερο 9. Τέλος όταν περάσουν 8 κύκλοι από την προηγούμενα επιλογή επιλέγονται οι έκτες είσοδοι των πολυπλεκτών έτσι ώστε να υπολογιστεί το συνολικό άθροισμα. Σε αυτή την περίπτωση λόγω του ότι τα αποτελέσματα που θα προστεθούν εξέρχονται από τον αθροιστή με δύο κύκλους διαφορά ο συγχρονισμός τους γίνεται μέσω των καταχωρητών νούμερο 9 και 10.



Σχήμα 4.5.4: Στο σχήμα αυτό παρουσιάζεται το ολοκληρωμένο Datapath που σχεδιάστηκε για την υλοποίηση τις σχέσης $Dnn < D_0$. Το "συννεφάκι" υποδεικνύει την λογική που χρησιμοποιείται για την υλοποίηση της απόλυτης τιμής.

Όπως φαίνεται στο σχήμα 4.5.3 μετά τον διαιρέτη ακολουθεί ένας συγκριτής ο οποίος υλοποιεί την σύγκριση $Dnn < D_0$, έχοντας ως σταθερή είσοδο B την τιμή 3 (D_0) σε αναπαράσταση floating point.

4.6 Μονάδα υπολογισμού και ελέγχου του κριτηρίου επιλογής του καλύτερου AFP από τα υποψήφια για επέκταση του μονοπατιού ομοιότητας

Ο υπολογισμός και ο έλεγχος του κριτηρίου επιλογής του καλύτερου AFP από τα υποψήφια για επέκταση του μονοπατιού ομοιότητας γίνεται χρησιμοποιώντας την σχέση 3.2.7 η οποία επαναλαμβάνεται πιο κάτω:

$$\frac{1}{n-1}\sum_{i=0}^{n-1} D_{in} < D_1$$

Στην περίπτωση της σχέσης 3.2.7 για τον υπολογισμό του D_{in} χρησιμοποιείται ο τύπος 3.2.2:

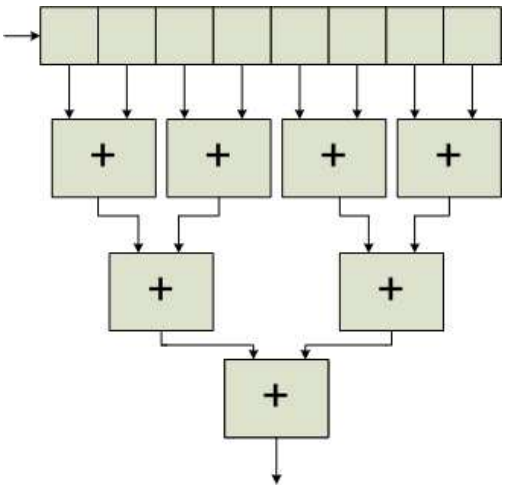
$$D_{ij} = \frac{1}{m}\left(\left| d^A_{p_i^A p_j^A} - d^B_{p_i^B p_j^B} \right| + \left| d^A_{p_i^A+m-1, p_j^A+m-1} - d^B_{p_i^B+m-1, p_j^B+m-1} \right| + \sum_{k=1}^{m-2} \left| d^A_{p_i^A+k, p_j^A+m-1-k} - d^B_{p_i^B+k, p_j^B+m-1-k} \right| \right)$$

Όπως φαίνεται από τον τύπο 3.2.2 για την υλοποίηση του χρειάζονται και πάλι, όπως και στην περίπτωση υπολογισμού του D_{nn}, δύο μονάδες υπολογισμού του $d_{i,j}$, και ένας αφαιρέτης που να υπολογίζει την διαφορά τους, αλλά και μία λογική έτσι ώστε το αποτέλεσμα του αφαιρέτη να είναι πάντα η απόλυτη τιμή της διαφοράς. Επίσης χρειάζεται μία διάταξη αθροιστών για την υλοποίηση των αθροίσματος των 8 τιμών και ένας διαιρέτης για την διαίρεση του αθροίσματος με το m.

Η διάταξη που χρησιμοποιείται για την υλοποίηση του αθροίσματος που παρουσιάζεται στον τύπο 3.2.2. διαφέρει από αυτή που σχεδιάστηκε για την περίπτωση του D_{nn}. Η διάταξη που θα σχεδιαστεί πρέπει να χρησιμοποιεί αθροιστή με latency 8 όπως αποφασίστηκε στην ενότητα 4.5 , να αθροίζει ομάδες των 8 στοιχείων αλλά να έχει την δυνατότητα να εκμεταλλευτεί την δυνατότητα για pipelining που παρέχεται από την μονάδες που χρησιμοποιούνται για την υλοποίηση, έτσι ώστε μόλις αθροίζονται τα 8 στοιχεία να υπάρχει η δυνατότητα να αθροιστούν τα επόμενα

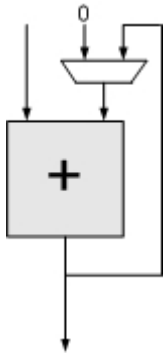
8 χωρίς να χρειάζεται να καθυστερήσουν σε κάποιο βήμα λόγω του ότι θα είναι απασχολημένες οι μονάδες που υλοποιούν το άθροισμα των 8 στοιχείων.

Ο συνδυασμός των 8 στοιχείων που πρέπει να προστεθούν μεταξύ τους και της επιλογής για latency 8 στον αθροιστή δεν επιτρέπει την χρήση αθροιστή με ανατροφοδότηση. Έτσι υλοποιήθηκε η λύση με τους εφτά αθροιστές που φαίνεται στο σχήμα 4.6.1.



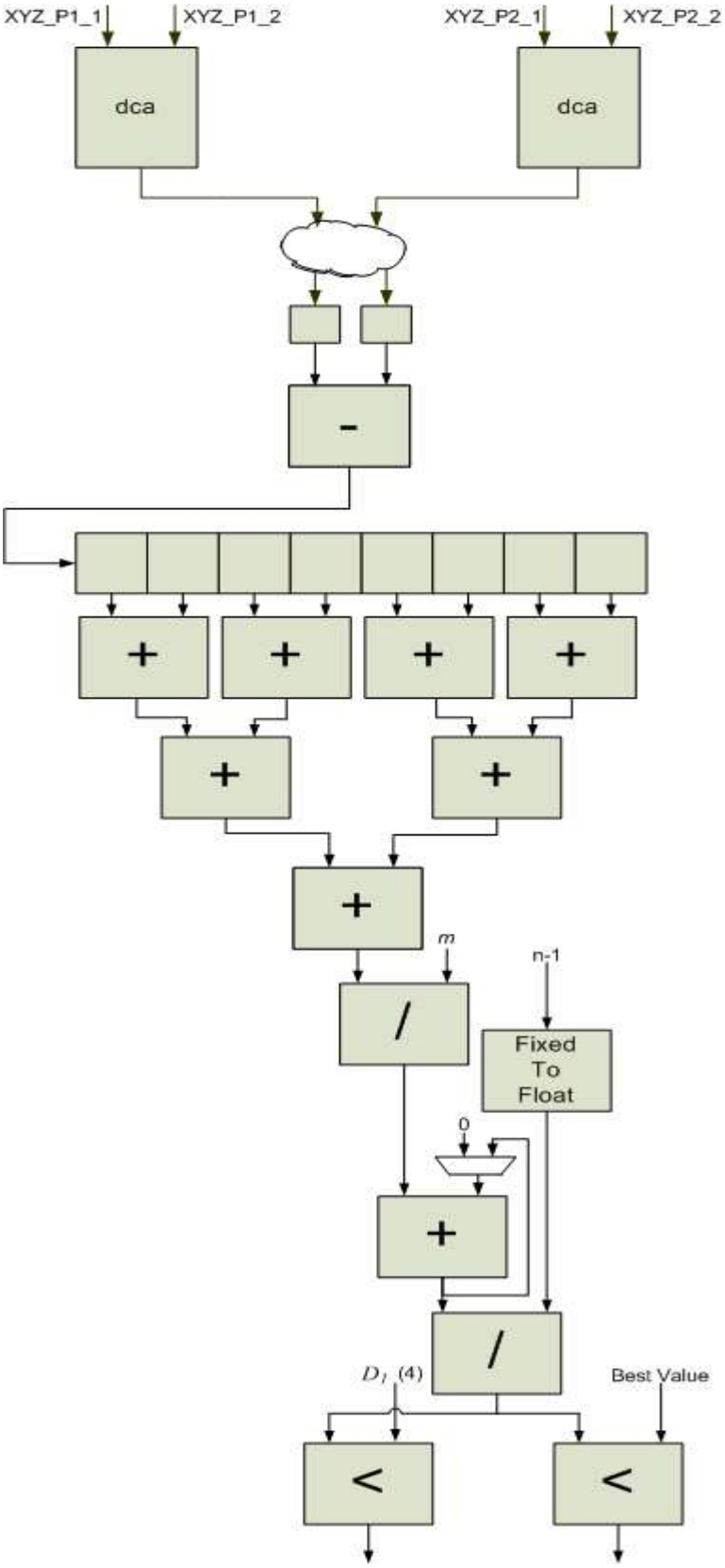
Σχήμα 4.6.1: Διάταξη εφτά αθροιστών με latency 8 για την άθροιση 8 στοιχείων. Στην σχεδίαση χρησιμοποιήθηκαν και οχτώ καταχωρητές για να συγχρονίσουν τα δεδομένα που αθροίζονται.

Τα *n* αθροίσματα των *n* οκτάδων, διαιρεμένα το καθένα με το *m*, πρέπει και αυτά να προστεθούν μεταξύ τους ($\sum_{i=0}^{n-1}$). Τα στοιχεία κάθε οκτάδας έρχονται το ένα μετά το άλλο σε κάθε κύκλο και μετά το όγδοο ακολουθούν τα επόμενα οκτώ. Έτσι μια οκτάδα αφού αθροιστεί η επόμενη αθροίζεται μετά από 8 κύκλους, και έτσι τα αθροίσματα των οκτάδων, διαιρεμένα με το *m*, δίνονται κάθε 8 κύκλους στον αθροιστή που πρέπει να τα προσθέσει. Αυτή η περίπτωση είναι ιδανική για την περίπτωση του ανατροφοδοτούμενου αθροιστή με latency 8 του σχήματος 4.6.2.



Σχήμα 4.6.2: Άθροιση των *n οκτάδων χρησιμοποιώντας* ανατροφοδότηση του αθροιστή με latency 8.

Στην ανατροφοδοτούμενη είσοδο του αθροιστή του σχήματος 4.6.2 είναι απαραίτητη η χρήση ενός πολυπλέκτη, έτσι όταν τελειώνει το άθροισμα των *n* στοιχείων (οκτάδων) η είσοδος αυτή να μηδενίζεται για να αθροιστούν τα επόμενα *n* στοιχεία. Η έξοδος του ανατροφοδοτούμενου αθροιστή διαιρείται μέσω ενός διαιρέτη και το αποτέλεσμα του διαιρέτη αποτελεί είσοδο για δύο συγκριτές. Ο ένας από αυτούς τους συγκριτές εξετάζει αν το αποτέλεσμα είναι μικρότερου του κατωφλίου D_1 (4), ενώ ο άλλος συγκριτής εξετάζει αν αυτή η τιμή είναι μικρότερη από την καλύτερη (άρα και μικρότερη από τις υπόλοιπες) τιμή από τις υποψήφιες n-άδες που προηγήθηκαν.

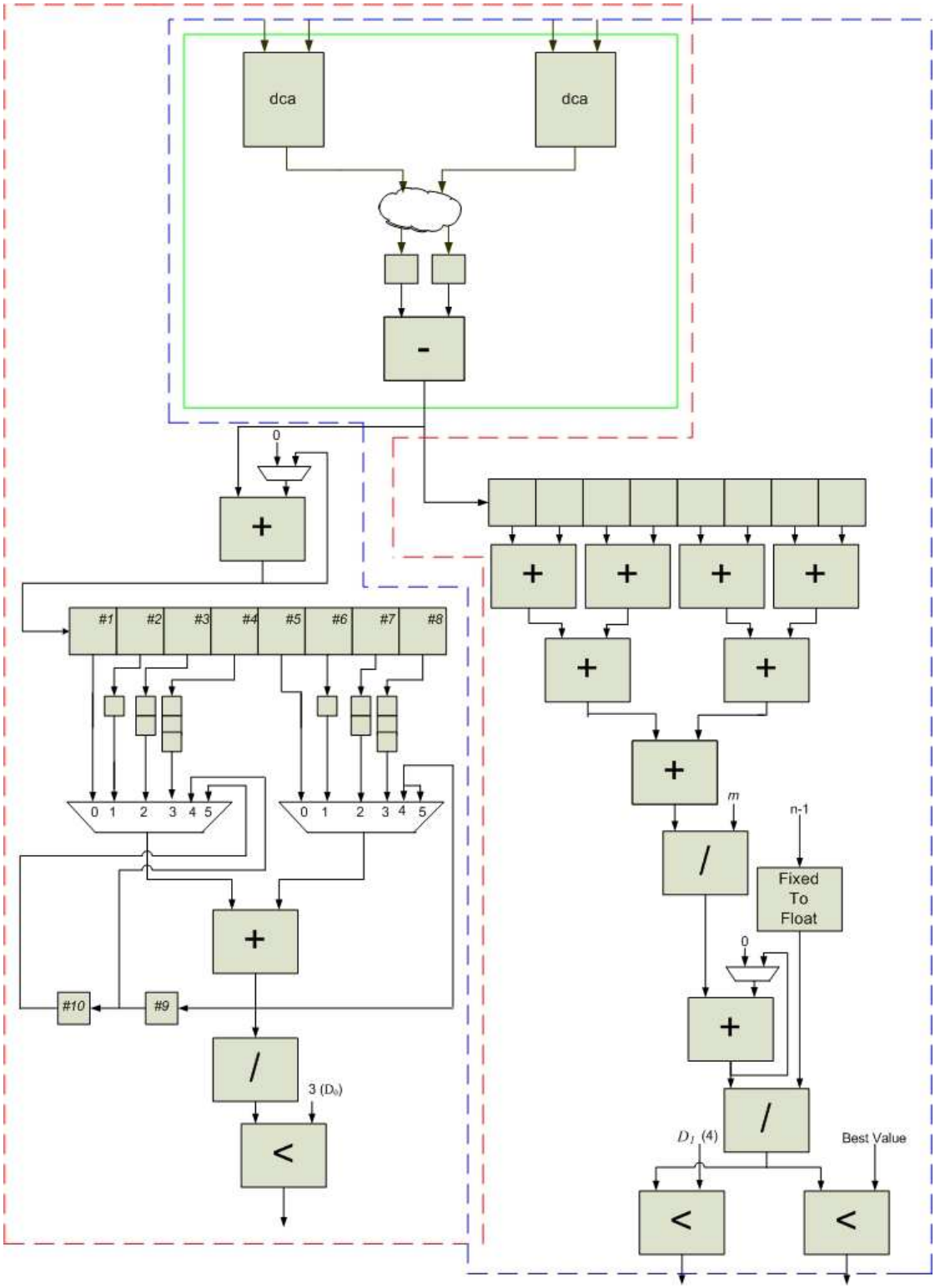


Σχήμα 4.6.3: Το ολοκληρωμένο Datapath της μονάδας υπολογισμού και ελέγχου του κριτηρίου επιλογής του καλύτερου AFP από τα υποψήφια για επέκταση του μονοπατιού ομοιότητας. Το "συννεφάκι" υποδεικνύει την λογική που χρησιμοποιείται για την υλοποίηση της απόλυτης τιμής

Όπως παρατηρείται στο σχήμα 4.6.3 η τιμή (n-1) πριν φτάσει ως είσοδος στον διαιρέτη περνάει από μία μονάδα μετατροπής αριθμών σταθερής υποδιαστολής σε αριθμούς κινητής υποδιαστολής έτσι ώστε να μετατραπεί στην αριθμητική αναπαράσταση στην οποία γίνονται όλες οι πράξεις της σχεδίασης. Έτσι η τιμή (n-1) εξέρχεται από την μονάδα ελέγχου, που θα παρουσιαστεί στην συνέχεια, σε μορφή ακεραίου 32-bit και η μονάδα fixed-to-float την δέχεται ως αριθμό σταθερής υποδιαστολής με ακέραιο μέρος 32-bit και την μετατρέπει σε αριθμό κινητής υποδιαστολής απλής ακρίβειας.

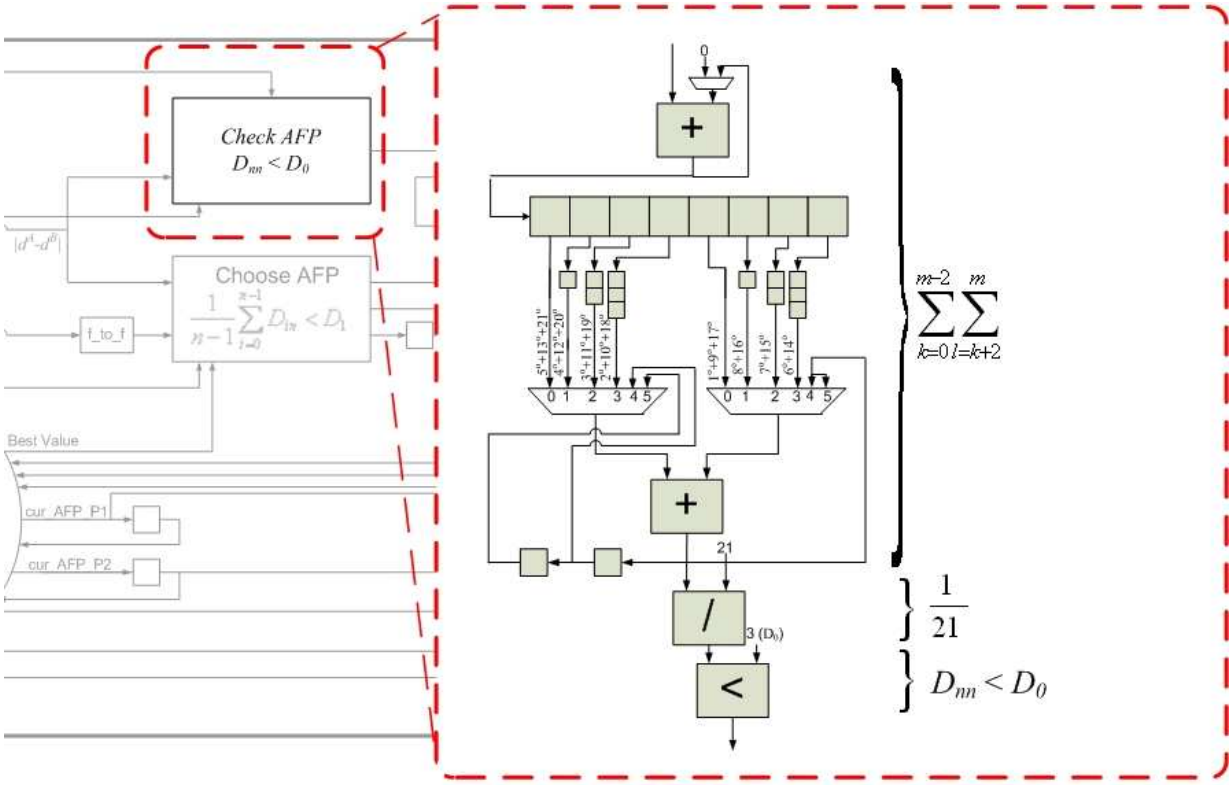
4.7 Βελτίωση της σχεδίασης για εξοικονόμηση των πόρων που έχουν κοινή χρήση από διαφορετικές μονάδες

Στις μονάδες που σχεδιάστηκαν στις ενότητες 4.5 και 4.6 παρατηρείται η χρήση ενός μέρους των datapath των δύο σχεδιάσεων είναι κοινό και στις δύο περιπτώσεις. Το κομμάτι αυτό αφορά των υπολογισμό της απόλυτης τιμής της διαφοράς των αποστάσεων $d_{i,j}$ και περιλαμβάνει δύο μονάδες υπολογισμού ευκλείδειας απόστασης, ένα αφαιρέτη και την λογική για την υλοποίηση της απόλυτης τιμής. Έτσι για εξοικονόμηση πόρων δημιουργήθηκε μία νέα σχεδίαση που συνδυάζει τις δύο μονάδες των σχημάτων 4.5.4 και 4.6.3 σε μία, όπου το κοινό τους μέρος υπάρχει στην σχεδίαση μόνο μία φορά.

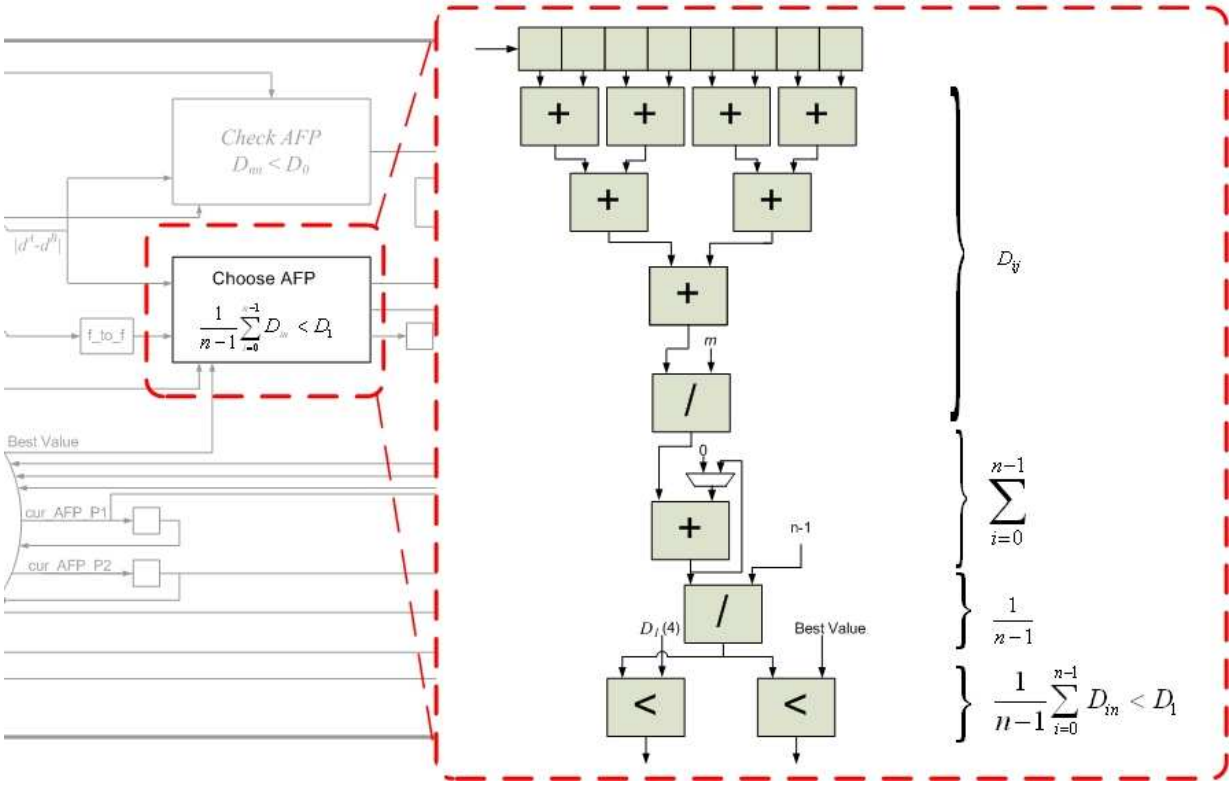


Σχήμα 4.7: Στο πιο πάνω σχήμα παρουσιάζεται η αλλαγή που έγινε στην σχεδίαση με στόχο την εξοικονόμηση πόρων. Μέσα στο πράσινο πλαίσιο περιλαμβάνεται η μονάδα Distance differential. Το μονοπάτι που περικλείεται από την κόκκινη διακεκομμένη χρησιμοποιείται για τον υπολογισμό της $Dnn < D_0$, ενώ η μπλε διακεκομμένη γραμμή δείχνει το μονοπάτι μέσω του οποίου υπολογίζεται η

$$\frac{1}{n-1}\sum_{i=0}^{n-1} D_{in} < D_1 .$$



Σχήμα 4.8: Στο πιο πάνω σχήμα παρουσιάζεται η μονάδα υπολογισμού του Dnn και σύγκρισής του με το D_0, όπως διαφοροποιήθηκε κατά την βελτίωση της σχεδίασης για εξοικονόμηση πόρων.



Σχήμα 4.9: Στο πιο πάνω σχήμα παρουσιάζεται η μονάδα υπολογισμού του κριτηρίου επιλογής του καλύτερου AFP, όπως διαφοροποιήθηκε κατά την βελτίωση της σχεδίασης για εξοικονόμηση πόρων.

4.8 Περιγραφή της μονάδας Fill Dnn memory, για την εγγραφή της μνήμης Dnn

Η εγγραφή των αποτελεσμάτων της σχέσης $Dnn < D_0$ γίνεται μέσω της μονάδας Fill Dnn memory. Αφού συμπληρωθεί ένα διάνυσμα από bits το οποίο αντιστοιχεί σε μία θέση μνήμης εγγράφεται στην μνήμη και στην συνέχεια συμπληρώνεται ξανά το διάνυσμα αυτό με την πληροφορία που αντιστοιχεί στην επόμενη θέση μνήμης έτσι ώστε όταν συμπληρωθεί να εγγραφεί στην θέση αυτή. Για την συμπλήρωση του διανύσματος αυτού ελέγχεται το αποτέλεσμα από της μονάδα υλοποίησης της σχέσης $Dnn < D_0$ και αν είναι 1 συμπληρώνεται η κατάλληλη θέση του διανύσματος με 1 αλλιώς παραμένει 0.

4.9 Περιγραφή της μονάδας Select the Bit, για την επιλογή του bit της μνήμης Dnn που αντιστοιχεί σε συγκεκριμένο AFP

Για να μπορέσει ένα AFP να αποτελέσει μέρος κάποιου μονοπατιού ομοιότητας πρέπει να γίνει ο έλεγχος του bit που αντιστοιχεί στην μνήμη Dnn γι' αυτό το AFP. Για το διάβασμα και την επιλογή αυτού του bit δίνεται η θέση της διεύθυνσης που πρέπει να διαβαστεί και που αντιστοιχεί που αντιστοιχεί στην αρχική θέση του AFP στην πρωτεΐνη A. Στην συνέχεια εμπλέκεται και η μονάδα Select the Bit που αποτελεί μια μονάδα η οποία δέχεται ως εισόδους το περιεχόμενο της θέσης της μνήμης που έχει διαβαστεί και ένα αριθμό που αντιστοιχεί στην αρχική θέση του AFP στην πρωτεΐνη B. Στην μονάδα αυτή επιλέγεται από το περιεχόμενο της θέσης μνήμης που διαβάστηκε προηγουμένως το bit που αντιστοιχεί στον αριθμό αυτό. Έτσι γίνεται η επιλογή του bit που αντιστοιχεί στο συγκεκριμένο AFP.

4.10 Μονάδα ελέγχου

Η λειτουργία των μονάδων που απαρτίζουν το datapath ρυθμίζεται μέσω της μονάδας ελέγχου. Η μονάδα αυτή υλοποιείται ως μία μηχανή πεπερασμένων καταστάσεων με την οποία καθορίζονται οι τιμές των σημάτων που αφορούν την

λειτουργία των μνημών, των μονάδων εκτέλεσης των αριθμητικών πράξεων και των πολυπλεκτών. Για ευκολότερη κατανόηση της λειτουργίας της μονάδας ελέγχου παρουσιάζεται ως μια ιεραρχική μηχανή πεπερασμένων καταστάσεων όπου κάποιες καταστάσεις μπορούν να ομαδοποιηθούν ως ξεχωριστές μηχανές πεπερασμένων καταστάσεων, οι οποίες οργανωμένες ιεραρχικά αποτελούν μία μηχανή πεπερασμένων καταστάσεων.

Η αρχική κατάσταση της μηχανής πεπερασμένων καταστάσεων είναι η κατάσταση RESET. Η μηχανή πεπερασμένων καταστάσεων βρίσκεται στην κατάσταση RESET όταν το σήμα Reset της σχεδίασης έχει πάρει την τιμή 1. Στην κατάσταση αυτή αρχικοποιούνται όλα τα σήματα της μονάδας ελέγχου στην τιμή 0 ή σε κάποια άλλη τιμή σε όσες περιπτώσεις χρειάζεται. Η κατάσταση στη οποία μεταβαίνει η μηχανή στον επόμενο κύκλο είναι η FirstSum_Dij.

4.10.1 Πρώτη φάση λειτουργίας

Αρχικά η μέθοδος που ακολουθήθηκε για την υλοποίηση απαιτεί τον έλεγχο του κριτηρίου $Dnn < D_0$, (3.2.1) για κάθε πιθανό AFP που μπορεί να υπάρξει από τις δύο πρωτεΐνες. Σε αυτή τη φάση συμπληρώνεται η μνήμη Dnn με 1 στο bit που αντιστοιχεί σε ένα AFP όταν ισχύει γι' αυτό το κριτήριο 3.2.1 και με 0 όταν δεν ισχύει.

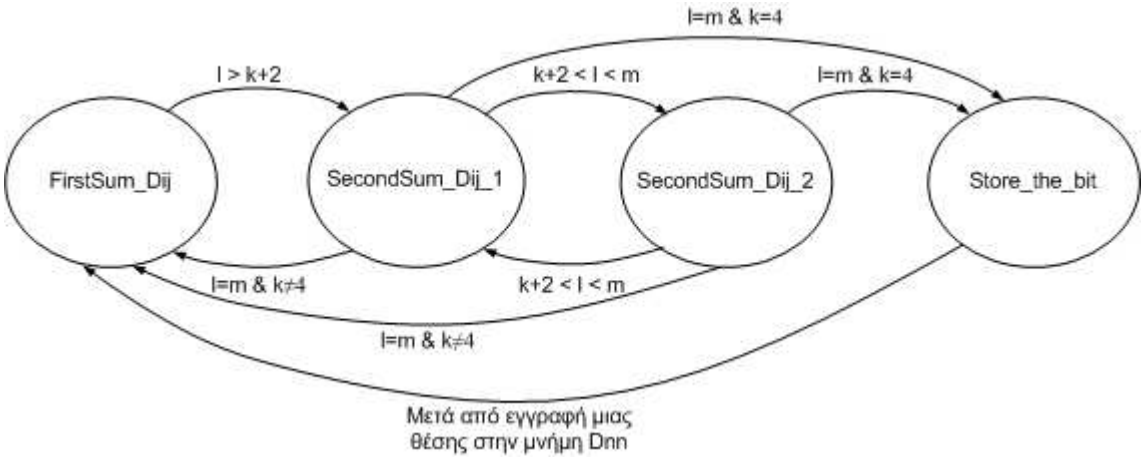
Η διαδικασία αυτή, η οποία αφορά την $Dij = \frac{1}{21}\left(\sum_{k=0}^{m-2}\sum_{l=k+2}^{m}\left|d^{A}_{p_i^A+k,p_j^A+l} - d^{B}_{p_i^B+k,p_j^B+l}\right|\right)$ (3.3.1) ελέγχεται μέσω των καταστάσεων FirstSum_Dij SecondSum_Dij_1, SecondSum_Dij_2 και Store_the_bit. Στις καταστάσεις αυτές διευθυνσιοδοτούνται σε κάθε κύκλο οι μνήμες MEM_Protein1 και MEM_Protein2. Οι διευθύνσεις αυτές υπολογίζονται με την χρήση των τιμών k και l. Ο καθορισμός της τιμής του k για το $\sum_{k=0}^{m-2}$, αλλά και του l για την πρώτη περίπτωση του $\sum_{l=k+2}^{m}$, δηλαδή όταν το l=k+2, γίνεται στη κατάσταση FirstSum_Dij. Η επόμενη κατάσταση που μεταβαίνει η μηχανή μετά την FirstSum_Dij είναι η SecondSum_Dij_1.

Οι καταστάσεις SecondSum_Dij_1 και SecondSum_Dij_2 καθορίζουν και τις υπόλοιπες τιμές για το l. Όσον αφορά την αλλαγή των καταστάσεων γίνεται μία

μετάβαση από την SecondSum_Dij_1 στην SecondSum_Dij_2 και από την SecondSum_Dij_2 στην SecondSum_Dij_1 μέχρι να ολοκληρωθεί η διαδικασία του αθροίσματος $\sum_{l=k+2}^{m}$, δηλαδή μέχρι να δοθεί και η τιμή l=m. Στην περίπτωση αυτή εάν ισχύει και ότι k=4, τότε υπάρχει μετάβαση στην Store_the_bit αλλιώς γίνεται μετάβαση στην FirstSum_Dij. Στις καταστάσεις SecondSum_Dij_1 στην SecondSum_Dij_2 ελέγχεται επίσης και η αλλαγή του σήματος ελέγχου επιλογής, όταν αυτό χρειάζεται, των πολυπλεκτών 6 προς 1 της διάταξης του σχήματος 4.5.4 στις περιπτώσεις που αυτό πρέπει να πάρει κάποια άλλη τιμή εκτός της 0. Επίσης όσο αφορά τον πολυπλέκτη 2 προς 1 της ίδια διάταξης ο έλεγχός αν το σήμα ελέγχου επιλογής του πολυπλέκτη πρέπει να γίνει από 0 σε 1 γίνεται και στις τέσσερις αυτές καταστάσεις.

Στην κατάσταση Store_the_bit καθορίζει το επόμενο AFP, δηλαδή τις τιμές των p_i^A, p_j^A, p_i^B και p_j^B για τις οποίες θα επαναληφθεί η διαδικασία. Επίσης ενεργοποιείται το σήμα που “ειδοποιεί” την μονάδα Fill_Dnn memory για να ελέγξει αν πρέπει να αποθηκευτεί το αποτέλεσμα στο διάνυσμα και όταν έχουν ελεγχθεί τα n_B-8 AFPs για μία τιμή του p^A, ενεργοποιείται και το σήμα για την εγγραφή του διανύσματος στην μνήμη Dnn. Από αυτή την κατάσταση γίνεται συνεχώς μετάβαση στην FirstSum_Dij εκτός από την περίπτωση όπου έχει ολοκληρωθεί η διαδικασία συμπλήρωσης της μνήμης Dnn όπου και γίνεται μετάβαση στην First_AFP.

Η διαδικασία αυτή γίνεται με πλήρη χρήση του pipelining για το λόγο αυτό δεν γίνεται σε κάποια συγκεκριμένη από τις τέσσερις καταστάσεις, αλλά σε όλες, ο έλεγχος που αφορά το σήμα ελέγχου επιλογής του πολυπλέκτη 2 προς 1 αφού ταυτόχρονα εκτελείται η διαδικασία για πολλές περιπτώσεις.



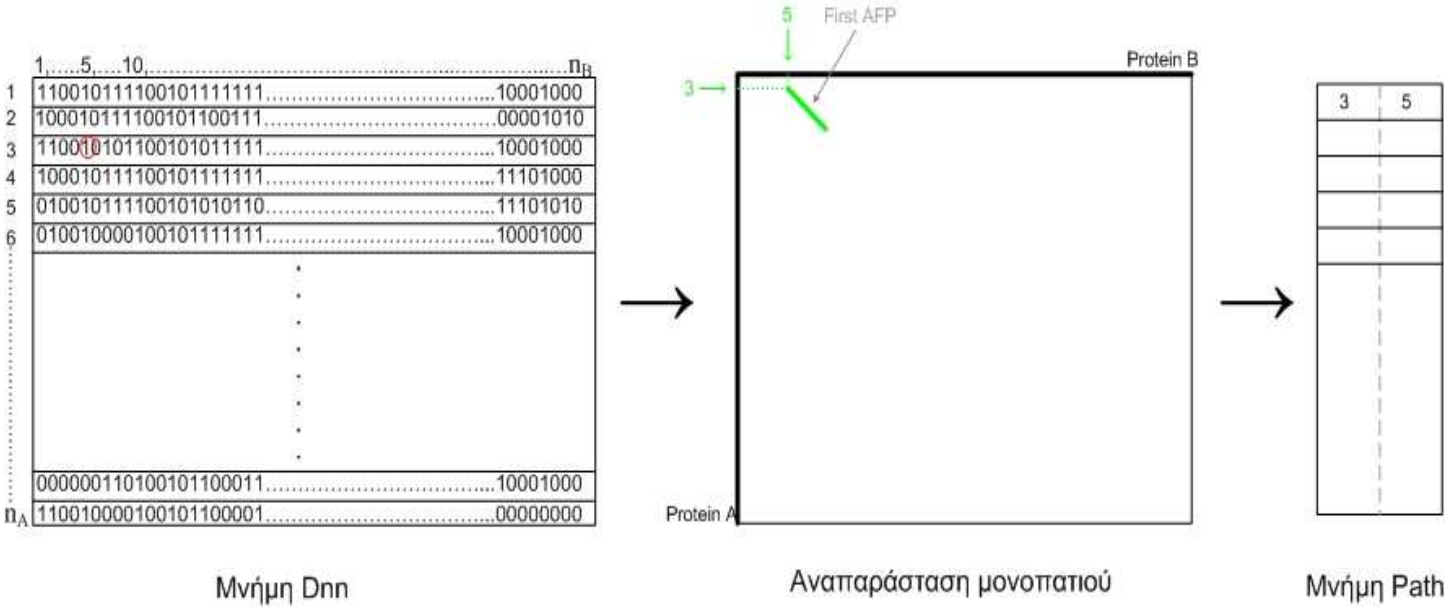
Σχήμα 4.10.1.1: Οι καταστάσεις που παρουσιάζονται στο σχήμα αυτό μπορούν να θεωρηθούν ως μία από τις μονάδες της ιεραρχικής μηχανής πεπερασμένων καταστάσεων αφού και οι τέσσερις αφορούν την διαδικασία εγγραφής της μνήμης Dnn.

4.10.2 Δεύτερη φάση λειτουργίας

Στην δεύτερη φάση λειτουργίας εκτελείται η διαδικασία υπολογισμού όλων των μονοπατιών ομοιότητας που μπορούν να δημιουργηθούν από την σύγκριση δύο πρωτεϊνών με στόχο την εύρεση του μεγαλύτερου μονοπατιού.

Η διαδικασία δημιουργίας ενός μονοπατιού αρχίζει στην κατάσταση First_AFP με τον καθορισμό του αρχικού AFP. Αν έχουν ήδη επιλεγεί όλα τα AFPs ως αρχικά τότε η διαδικασία τερματίζει και η μηχανή μεταβαίνει στην κατάσταση Final_State, αλλιώς μεταβαίνει στην Wait1 έτσι ώστε να περάσει ένας κύκλος για να διαβαστεί από την μνήμη Dnn το bit που αντιστοιχεί στο αρχικό AFP. Στον επόμενο κύκλο στην κατάσταση Check_First_AFP αν το bit είναι 1 τότε γράφονται στην μνήμη Path οι αρχικές θέσης του AFP αφού αποτελεί το πρώτο AFP του μονοπατιού και στον επόμενο κύκλο γίνεται μετάβαση στην Check_AFPs1, ενώ στην περίπτωση που είναι 0 τότε γίνεται μετάβαση στην First_AFP.

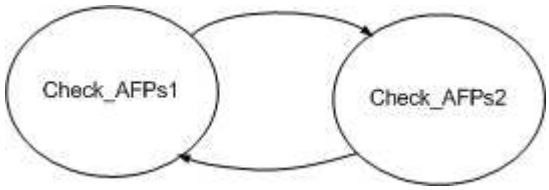
Σχήμα 4.10.2.1: Ως το δεύτερο μέρος της ιεραρχικής μηχανής πεπερασμένων καταστάσεων μπορούν να θεωρηθούν οι καταστάσεις στις οποίες ελέγχεται αν ένα AFP μπορεί να χρησιμοποιηθεί στο μονοπάτι ως το αρχικό AFP του μονοπατιού.



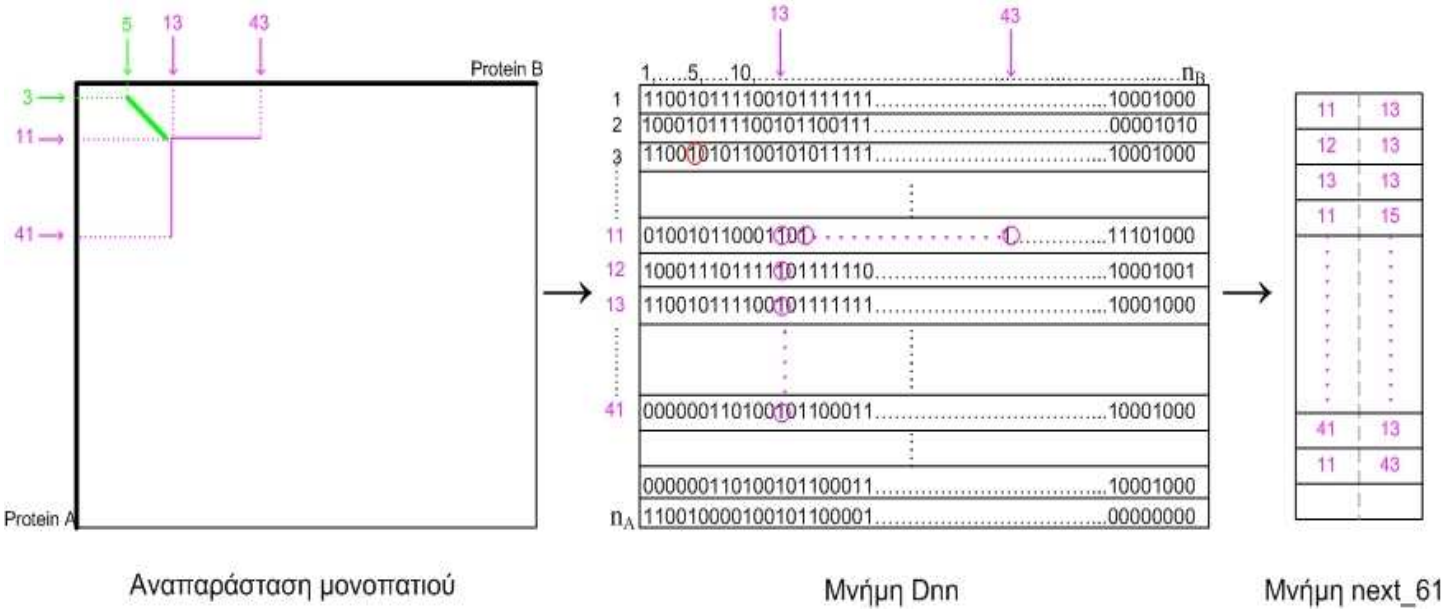
Σχήμα 4.10.2.2: Εδώ δίνεται μία σχηματική επεξήγηση για την διαδικασία που ελέγχεται μέσω των καταστάσεων First_AFP, Wait1 και Check_First_AFP. Στην περίπτωση του σχήματος το AFP που έχει ως αρχικές θέσεις στις πρωτεΐνες A και B, τις θέσεις 3 και 5 αντίστοιχα, μπορεί να θεωρηθεί ως το αρχικό AFP του μονοπατιού αφού το bit που του αντιστοιχεί στην μνήμη Dnn είναι 1. Έτσι γράφεται στην μνήμη Path ως το πρώτο AFP του μονοπατιού που υπολογίζεται την συγκεκριμένη στιγμή.

Στις καταστάσεις Check_AFPs1 και Check_AFPs2 διαβάζονται και ελέγχονται τα bits που αντιστοιχούν στα 61 υποψήφια AFPs για να προστεθούν στο μονοπάτι μετά το τελευταίο που προστέθηκε. Για όσα από αυτά η τιμή στην Dnn είναι 1 γράφονται στην μνήμη next_61 οι αρχικές τους θέσεις. Όσο διαρκεί η διαδικασία αυτή οι μεταβάσεις των καταστάσεων γίνονται μεταξύ των Check_AFPs1

και Check_AFPs2. Όταν ολοκληρωθεί η διαδικασία αν κανένα από τα 61 υποψήφια AFPs δεν έχει γραφτεί στην next_61 αυτό σημαίνει ότι δεν μπορεί να συνεχίσει το μονοπάτι και αρχίζει η διαδικασία δημιουργίας του επόμενου μονοπατιού μεταβαίνοντας στην First_AFP. Στην αντίθετη περίπτωση η διαδικασία συνεχίζεται μεταβαίνοντας στην κατάσταση sum_Din_calc1.



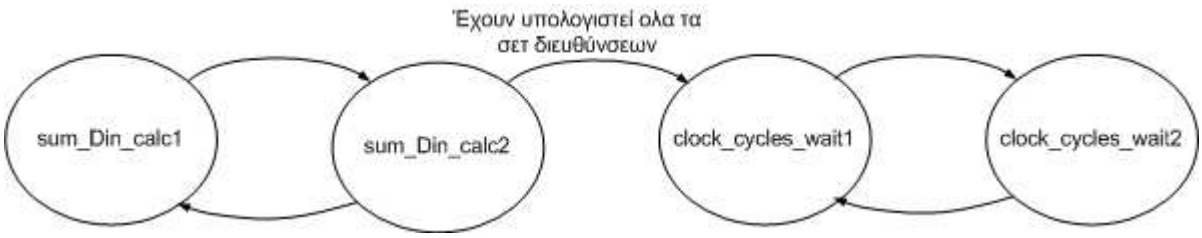
Σχήμα 4.10.2.3: Οι δύο καταστάσεις στις οποίες εξετάζεται η εγκυρότητα και η εγγραφή στην μνήμη next_61 των υποψήφιων, προς επέκταση του μονοπατιού, AFPs μπορούν να θεωρηθούν ως το τρίτο μέρος της ιεραρχικής μηχανής πεπερασμένων καταστάσεων.



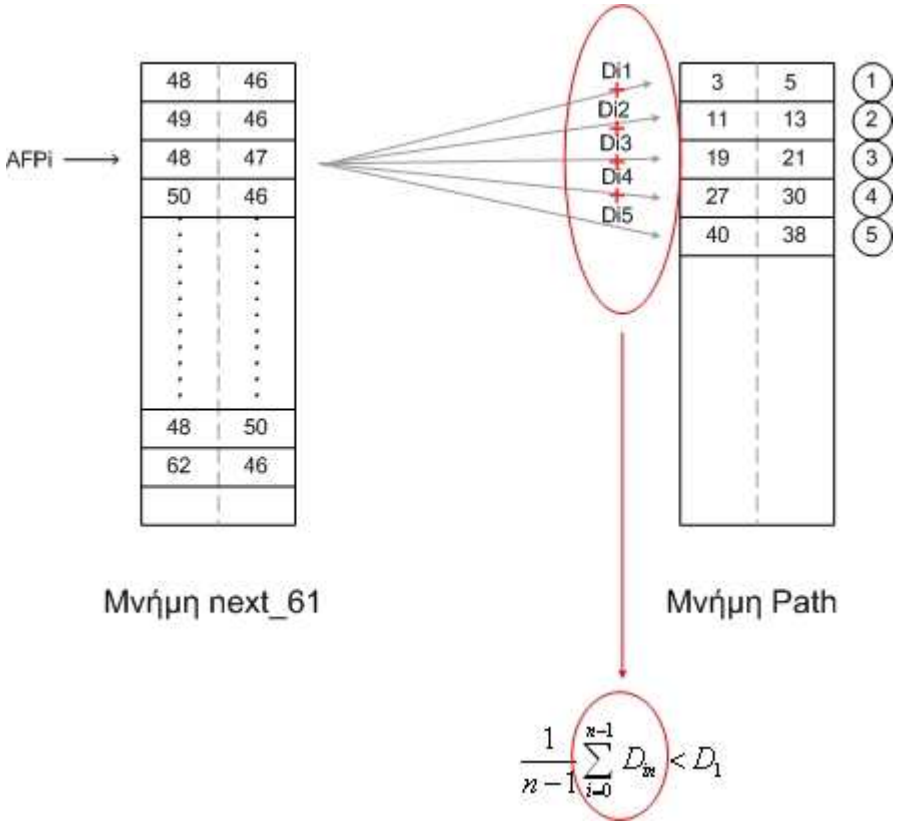
Σχήμα 4.10.2.4: Εδώ δίνεται μία σχηματική επεξήγηση της διαδικασίας ελέγχου των υποψήφιων προς επέκταση του μονοπατιού AFPs και οι εγγραφή των κατάλληλων, δηλαδή αυτών που το bit που τους αντιστοιχεί στην μνήμη Dnn είναι 1, στην μνήμη next_61.

Στις καταστάσεις sum_Din_calc1 και sum_Din_calc2 ελέγχεται η σχέση. $\frac{1}{n-1}\sum_{i=0}^{n-1} D_{in} < D_1$. Διαβάζονται όλα τα AFPs έχουν εγγραφεί στην μνήμη next_61 και

για κάθε ένα από αυτά υπολογίζεται η τιμή $\frac{1}{n-1}\sum_{i=0}^{n-1} D_{in}$, σε συνδυασμό με όλα τα AFPs που έχουν εγγραφεί στην Path και αποτελούν τα AFPs που έχουν ήδη προστεθεί στο συγκεκριμένο μονοπάτι. Αφού υπολογιστεί η τιμή αυτή γίνεται σύγκριση τόσο με το D_1 για να ελεγχθεί αν $\frac{1}{n-1}\sum_{i=0}^{n-1} D_{in} < D_1$, καθώς και με την μικρότερη τιμή από τα 61 που έχει υπολογιστεί μέχρι στιγμής έτσι ώστε να επιλεγεί το καλύτερο. Στις καταστάσεις αυτές στις οποίες μεταβαίνει η μηχανή εναλλάξ γίνεται ο καθορισμός των σημάτων που αφορούν την διευθυνσιοδότηση των μνημών MEM_Protein1 και MEM_Protein2, με την χρήση των δεδομένων που διαβάζονται από τις μνήμες next_61 και Path, οπόταν ελέγχεται και το διάβασμα των μνημών next_61 και Path. Επίσης γίνεται ο έλεγχος του σήματος για τον πολυπλέκτη 2 προς 1, καθορίζεται η τιμή Best_Value για τον συγκριτή, καθώς και η τιμή n-1 για την μονάδα Fixed To Float του σχήματος 4.6.3. Έτσι αν ένα AFP αποδειχθεί ως ιδανικότερο από τα προηγούμενα που ελέγχθηκαν για να προστεθεί στο μονοπάτι αποθηκεύεται σε ένα σήμα η θέση του στην μνήμη next_61 για να μπορέσει να χρησιμοποιηθεί αργότερα, ενώ η τιμή που υπολογίστηκε και μέχρι στιγμής αποτελεί την μικρότερη δίνεται στο σήμα Best_Value. Για κάθε AFP που έχει γραφτεί στην μνήμη next_61 υπολογίζονται 8*(n-1) σετ διευθύνσεων για τις μνήμες MEM_Protein1 και MEM_Protein2. Αφού υπολογιστούν και για όλα η μηχανή μεταβαίνει στην clock_cycles_wait1.



Σχήμα 4.10.2.5: Οι καταστάσεις στις οποίες υπολογίζεται η καταλληλότητα των AFPs και επιλέγεται το καλύτερο, από όσα πληρούν τα κριτήρια, μπορούν να θεωρηθούν ως το τέταρτο μέρος της ιεραρχικής μηχανής πεπερασμένων καταστάσεων.



Σχήμα 4.10.2.6: Εδώ δίνεται μία σχηματική επεξήγηση της διαδικασίας υπολογισμού του $\frac{1}{n-1}\sum_{i=0}^{n-1} D_{in}$ για ένα από τα AFPs που βρίσκονται στην μνήμη next_61.

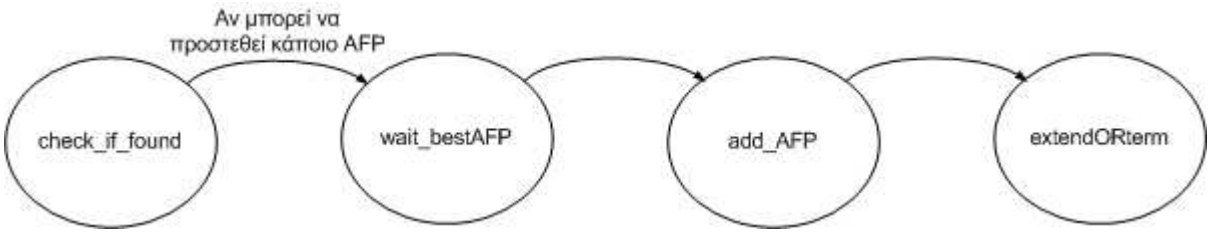
Στις καταστάσεις clock_cycles_wait1 και clock_cycles_wait2 συνεχίζεται η διαδικασία όπως και στις καταστάσεις sum_Din_calc1 και sum_Din_calc2, αλλά πλέον δεν συνεχίζεται το διάβασμα από τις μνήμες next_61 και Path καθώς και ο υπολογισμός των διευθύνσεων των μνημών MEM_Protein1 και MEM_Protein2. Ουσιαστικά στις καταστάσεις αυτές σταματάει η εκμετάλλευση της δυνατότητας που παρέχουν οι άλλες μονάδες για pipelining και το σύστημα περιμένει να υπολογιστεί και για το τελευταίο AFP, που διαβάστηκε από την μνήμη next_61, η τιμή $\frac{1}{n-1}\sum_{i=0}^{n-1} D_{in}$ και να γίνουν οι συγκρίσεις με το D_1 και το Best_Value. Αυτή η διαδικασία είναι απαραίτητη αφού για να συνεχίσει η επέκταση του μονοπατιού πρέπει να εξεταστούν όλα τα υποψήφια AFPs και να επιλεγεί το καλύτερο. Όταν δοθούν τα αποτελέσματα των συγκρίσεων και για το τελευταίο AFP γίνεται μετάβαση στην check_if_found. Εδώ γίνεται ένας έλεγχος και σε περίπτωση που κανένα από τα AFPs που διαβάστηκαν από την next_61 δεν ικανοποιεί το κριτήριο

$\frac{1}{n-1}\sum_{i=0}^{n-1} D_{in} < D_1$, τότε γίνεται μετάβαση στην First_AFP για την έναρξη της δημιουργίας του επόμενου μονοπατιού, αλλιώς γίνεται μετάβαση στην wait_bestAFP.

Η κατάσταση wait_bestAFP χρησιμεύει στο να καταναλωθεί ένας κύκλος έτσι ώστε να διαβαστεί από την μνήμη next_61 η θέση της διεύθυνσης που κρατήθηκε ως το ιδανικότερο AFP για να προστεθεί στο μονοπάτι. Η διαδικασία συνεχίζει στην add_AFP.

Στην κατάσταση add_AFP προστίθεται στο μονοπάτι το AFP που επιλέχθηκε, δηλαδή οι συντεταγμένες των αρχικών του θέσεων στις πρωτεΐνες που διαβάστηκαν από την μνήμη next_61, με την εγγραφή του στην μνήμη Path. Επίσης αυξάνεται σο σήμα που μετράει το μέγεθος του μονοπατιού κατά 1. Αν αυτό είναι το μεγαλύτερο μονοπάτι από όλα όσα έχουν εξεταστεί τότε σημειώνεται το μέγεθός του ως το βέλτιστο που επιτεύχθηκε μέχρι στιγμής. Την κατάσταση add_AFP ακολουθεί πάντα η extendORterm.

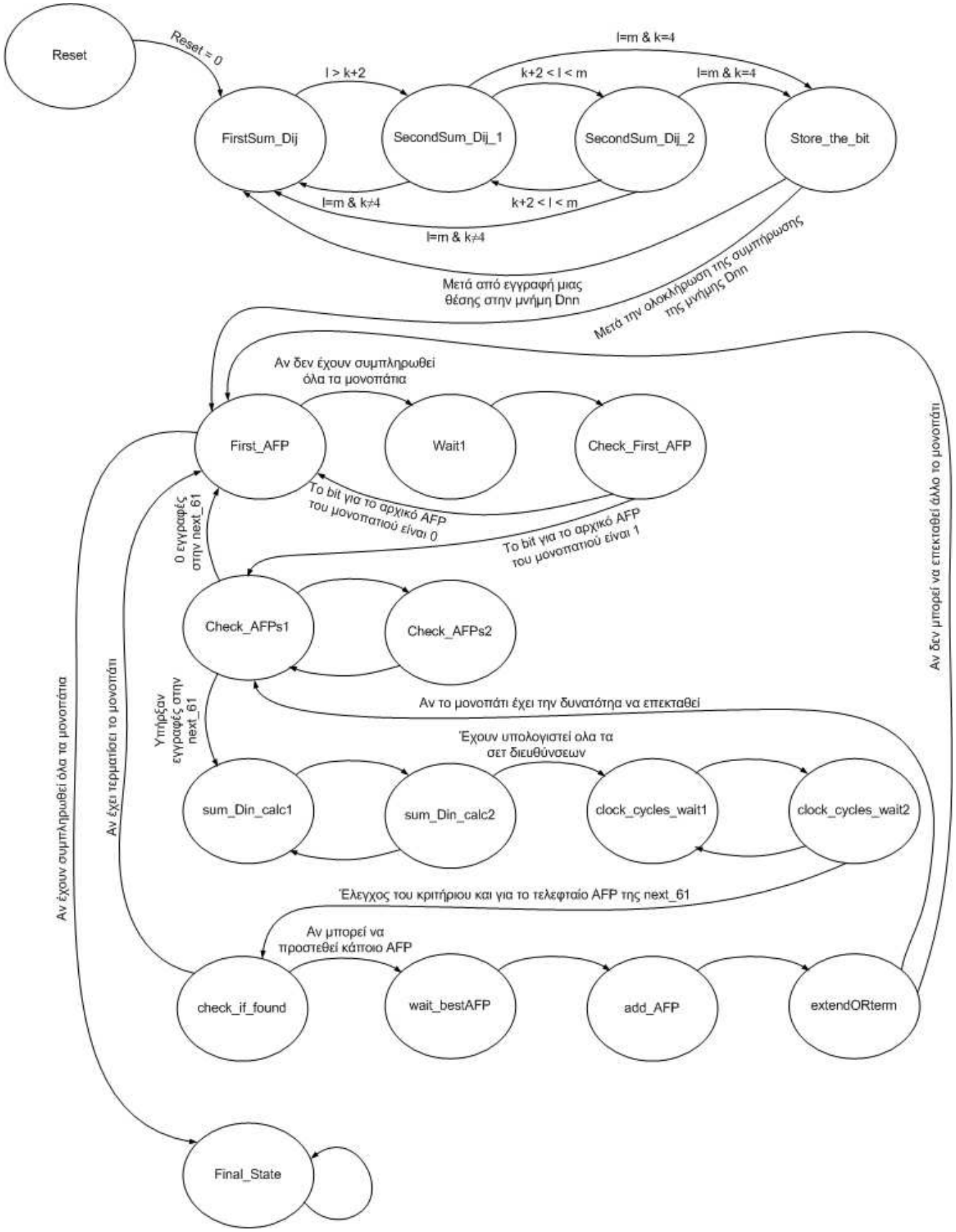
Ένα μονοπάτι όταν επεκταθεί τόσο που αν προστεθεί σε αυτό άλλο AFP ξεφεύγει από τα όρια που καθορίζει το μέγεθος των δύο πρωτεϊνών τότε το μονοπάτι αυτό έχει ολοκληρωθεί και τερματίζεται, αλλιώς συνεχίζει η διαδικασία επέκτασής του. Η περιπτώσεις αυτές ελέγχονται στην κατάσταση extendORterm. Έτσι αν ισχύει η πρώτη περίπτωση μεταβαίνει στην κατάσταση First_AFP, ενώ αν ισχύει η δεύτερη τότε μεταβαίνει στην Check_AFPs1 για να συνεχίσει η διαδικασία επέκτασης του μονοπατιού.



Σχήμα 4.10.2.7: Οι καταστάσεις στις οποίες εξετάζεται αν υπάρχει κάποιο AFP κατάλληλο για να προστεθεί στο μονοπάτι και αν υπάρχει προστίθεται σε αυτό, μπορούν να θεωρηθούν ως το πέμπτο μέρος της ιεραρχικής μηχανής πεπερασμένων καταστάσεων.

Όταν υπολογιστούν όλα τα μονοπάτια ομοιότητας τότε η μηχανή μεταβαίνει στην κατάσταση Final_State όπου ενεργοποιείται το σήμα Done το οποίο

σηματοδοτεί τον τερματισμό της όλης διαδικασίας. Η μηχανή μεταβαίνει συνεχώς σε αυτή την κατάσταση.



Σχήμα 4.10.2.8: Μια ολοκληρωμένη άποψη της μηχανής πεπερασμένων καταστάσεων με την οποία υλοποιείται η μονάδα ελέγχου.

5

Έλεγχος ορθής λειτουργίας και Αποτίμηση της απόδοσης

Στο παρών κεφάλαιο παρουσιάζεται η διαδικασία ταυτοποίησης των αποτελεσμάτων της εξόδου της αρχιτεκτονικής που υλοποιήθηκε, καθώς και τα αποτελέσματα απόδοσης σε σύγκριση με PC.

5.1 Έλεγχος ορθής λειτουργίας

Η προσομοίωση της λειτουργίας της αρχιτεκτονικής έγινε με το πρόγραμμα ModelSim SE 6.3f. Κατά την διαδικασία της υλοποίησης της αρχιτεκτονικής που σχεδιάστηκε, κάθε επιμέρους τμήμα της αρχιτεκτονικής προσομοιώθηκε ξεχωριστά και ελέγχθηκε η σωστή λειτουργία του.

Αφού ολοκληρώθηκε το σύστημα προσομοιώθηκε με στόχο την διαπίστωση της σωστής λειτουργίας του, χρησιμοποιώντας δέκα σετ εισόδων. Η σύγκριση των αποτελεσμάτων της προσομοίωσης έγινε και με τα αποτελέσματα του προγράμματος που υλοποιήθηκε κατά την διαδικασία της μοντελοποίησης, αλλά κυρίως έγινε με τα αποτελέσματα του προγράμματος που περιλαμβάνεται στο Bioperf το οποίο αποτελεί σημείο αναφοράς στην αξιολόγηση αρχιτεκτονικών υπολογιστών υψηλών επιδόσεων σε εφαρμογές βιοπληροφορικής. Στην περίπτωση του προγράμματος στο Bioperf χρησιμοποιείται και η σχέση 3.2.4 η που μπορεί να θεωρηθεί ως βελτιστοποίηση αφού απορρίπτει κάποια μονοπάτια κατά την διαδικασία σχηματισμού τους και έτσι αποτρέπεται ο άσκοπος υπολογισμός τους που αποτελεί και το βαρύτερο υπολογιστικά κομμάτι του αλγορίθμου, κάτι το οποίο δεν συμπεριλαμβάνεται στο σύστημα που υλοποιήθηκε στα πλαίσια αυτής της εργασίας. Επειδή αναμένεται ότι αυτή η μείωση του υπολογιστικού όγκου θα επηρεάζει σοβαρά την επίδοση του αλγορίθμου και εντονότερα σε περιπτώσεις όπου τα μονοπάτια ομοιότητας που δημιουργούνται από την σύγκριση των πρωτεϊνών είναι μεγάλα σε μήκος, εκτελέστηκαν τα ίδια πειράματα και με την εφαρμογή που υλοποιήθηκε στην διαδικασία μοντελοποίησης του αλγορίθμου στην οποία δεν συμπεριλαμβάνεται η

συγκεκριμένη βελτιστοποίηση. Αυτό έχει ως στόχο η σύγκριση των δύο περιπτώσεων να βοηθήσει στην ορθότερη εκτίμηση της επιτάχυνσης.

Λόγω του μεγάλου χρόνου που χρειάζεται η διαδικασία προσομοίωσης της λειτουργίας της αρχιτεκτονικής που υλοποιήθηκε, ενδεικτικά για την περίπτωση ενός ζεύγους πρωτεϊνών μεγέθους 98, με βέλτιστο μονοπάτι ομοιότητας 9 AFPs (72 κατάλοιπα), χρειάστηκαν τρεις μέρες, ο έλεγχος της λειτουργίας έγινε με πρωτεΐνες που η σύγκρισή τους δίνει βέλτιστο μονοπάτι ομοιότητας μικρότερο των 10 AFPs.

Τα πειράματα που διεξήχθησαν για τον έλεγχο της λειτουργίας και της απόδοσης παρουσιάζονται στον πιο κάτω πίνακα:

Είσοδοι (Μεγέθη)	Μήκος μονοπατιού σε AFPs (σε αριθμό κατάλοιπων)
1LYP-1OLG:A (32-42)	3 (24)
1BBT:4-1TMF:4 (85-45)	3 (24)
1TIG-3FIS:A (94-98)	5 (40)
1VIH-1PYT:A (71-94)	7 (56)
1KUM-1TUL (108-108)	8 (64)
3FIS:A-3FIS:B (98-98)	9 (72)
1SCE:A-1PUC (112-105)	12 (96)
1CPC:L-1COL:A (172-204)	15 (120)
1HBA:A-4HHB:B (141-146)	17 (136)
1ATP:E-1KOB:A (350-387)	33 (264)

Πίνακας 5.1: Τα πειράματα που διεξήχθησαν για τον έλεγχο της λειτουργίας και της απόδοσης.

5.2 Απόδοση συστήματος

Με την χρήση του Xilinx ISE 10.1 έγινε η διαδικασία Synthesis και Place & Route με την οποία προέκυψαν η συχνότητα ρολογιού και οι πόροι που καταναλώνει η αρχιτεκτονική που υλοποιήθηκε.

Clock Frequency : **275 MHz**.

Device Utilization Summary (xc5vsx240t-2ff1738)			
	Used	Available	Utilization
Number of Slice Registers	15786	149760	9%
Number of Slice LUTs	13059	149760	8%
Number of Slice LUT-FF pairs	18371	149760	12%
Number of bonded IOBs	87	960	9%
Number of Block RAM/FIFO	36	516	6%
Number of BUFG/BUFGCTRLs	16	32	50%
Number of DSP48Es	64	1056	6%

Πίνακας 5.2: Χρησιμοποίηση πόρων της FPGA για την υλοποίηση της αρχιτεκτονικής.

5.3 Αποτίμηση απόδοσης

Η εφαρμογή του αλγορίθμου CE που περιλαμβάνεται στο πακέτο Bioperf εκτελέστηκε σε PC με επεξεργαστή Intel Pentium 4 στα 2,66 GHz, με 1GB RAM και λειτουργικό σύστημα Ubuntu 7.10. Για την μέτρηση του χρόνου CPU κάθε πειράματος χρησιμοποιήθηκε το πρόγραμμα Vtune. Για κάθε πείραμα το πρόγραμμα για τον αλγόριθμο CE εκτελέστηκε 10 φορές και λήφθηκε ο μέσος όρος εκτέλεσης.

Είσοδοι (Μεγέθη)	Μήκος μονοπατιού σε AFPs (σε αριθμό κατάλοιπων)	Χρόνος εκτέλεσης CPU σε Pentium4@2,66GHz (secs)
1LYP-1OLG:A (32-42)	3 (24)	0,0113255
1BBT:4-1TMF:4 (85-45)	3 (24)	0,0077170
1TIG-3FIS:A (94-98)	5 (40)	0,0437860
1VIH-1PYT:A (71-94)	7 (56)	0,0320720
1KUM-1TUL (108-108)	8 (64)	0,0736660
3FIS:A-3FIS:B (98-98)	9 (72)	0,0207470
1SCE:A-1PUC (112-105)	12 (96)	0,0390880
1CPC:L-1COL:A (172-204)	15 (120)	0,5427210
1HBA:A-4HHB:B (141-146)	17 (136)	0,1248810
1ATP:E-1KOB:A (350-387)	33 (264)	2,0722638

Πίνακας 5.3.1: Χρόνοι εκτέλεσης των δέκα πειραμάτων σε Pentium4@2,66GHz.

Όπως αναφέρθηκε και προηγουμένως η προσομοίωση πειραμάτων με μεγάλες εισόδους και κυρίως με μεγάλα μονοπάτια ομοιότητας είναι μία πολύ χρονοβόρα διαδικασία και για τον λόγο αυτό οι περιπτώσεις αυτές δεν μπόρεσαν να προσομοιωθούν για να μετρηθούν οι κύκλοι ρολογιού για την εκτέλεση των πειραμάτων αυτών. Έτσι μετρήθηκε μέσω τις προσομοίωσης του συστήματος ο χρόνος εκτέλεσης μόνο για περιπτώσεις με μικρές εισόδους. Παρ' όλα αυτά για να υπάρχει μία πιο πλήρης άποψη για την αποτίμηση της απόδοσης υπολογίστηκε θεωρητικά ο χρόνος εκτέλεσης όλων των πειραμάτων.

Ο πιο κάτω μαθηματικός υπολογισμός δίνει μία εκτίμηση του αριθμού των κύκλων ρολογιού που χρειάζονται για να εκτελεστεί ένα πείραμα. Η εκτίμηση γίνεται με την χρήση κάποιων παραμέτρων που μετρήθηκαν με κάποιο επιπλέον κώδικα που προστέθηκε στην υλοποίηση του αλγορίθμου σε C που έγινε κατά την μοντελοποίηση. Αυτός ο επιπλέον κώδικας αφαιρέθηκε κατά την διαδικασία μέτρησης του χρόνου εκτέλεσης του προγράμματος.

Οι παράμετροι αυτοί αφορούν τα μεγέθη όλων των μονοπατιών ομοιότητας που δημιουργούνται από την σύγκριση των δύο πρωτεϊνών και τον αριθμό των AFPs που ελέγχονται κάθε φορά ως υποψήφια για να αποτελέσουν μέρος της επέκτασης του μονοπατιού ομοιότητας. Οι παράμετροι παρουσιάζονται στον υπολογισμό της 5.3.3 και είναι οι εξής:

i: το πλήθος για ένα συγκεκριμένο μέγεθος μονοπατιού. Οι τιμές που μπορεί να πάρει είναι από 0 μέχρι *((*n^A*-m)*(*n^B*-m).*

max_AFPs$_i$: το μήκος του μονοπατιού i σε αριθμό AFPs. Το πλήθος του συγκεκριμένου μήκους είναι το i.

AFPs_Dnn_1$_k$: ο αριθμός των υποψήφιων AFPs για τα οποία το bit που τους αντιστοιχεί στην μνήμη Dnn είναι 1 και για τα οποία θα υπολογιστεί το κριτήριο ελέγχου ομοιότητας με το οποίο επιλέγεται το βέλτιστο AFP για να προστεθεί στο μονοπάτι. Οι τιμές που μπορεί να πάρει είναι από 0 μέχρι 61.

Εκτίμηση αριθμού κύκλων που χρειάζονται για την συμπλήρωση της μνήμης Dnn:

$$Dnn_cycles = ((21*((n_A - 8)*(n_B - 8))) + 142) \quad (5.3.1)$$

Εκτίμηση αριθμού κύκλων για τον έλεγχο όλων των AFPs ως αρχικά AFPs του μονοπατιού:

$$CheckAllFirst_AFPs = ((n_A - 8)*(n_B - 8)*3) \quad (5.3.2)$$

Εκτίμηση αριθμού κύκλων για τον υπολογισμό όλων των μονοπατιών:

$$All_Paths = \sum_i \left(i * \sum_{k=0}^{\max_AFPs_i} \left(65 + \left(AFPs_Dnn_1_k * 8 * k\right) + 153\right) \right)$$

(5.3.3)

Σύνολο των εκτιμώμενων κύκλων:

$$Estimated_cycles = Dnn_cycles + CheckAllFirst_AFPs + All_Paths \quad (5.3.4)$$

Είσοδοι (Μεγέθη)	AFPs (κατάλοιπα)	Αριθμός κύκλων	Χρόνος εκτέλεσης (sec)	Εκτιμώμενος αριθμός κύκλων	Εκτιμώμενος χρόνος (sec)
1LYP-1OLG:A (32-42)	3 (24)	143423	0,0005215	132664	0,0004824
1BBT:4-1TMF:4 (85-45)	3 (24)	229367	0,0008341	139444	0,0005071
1TIG-3FIS:A (94-98)	5 (40)	3453869	0,0125595	3201324	0,0116412
1VIH-1PYT:A (71-94)	7 (56)	3135530	0,0114019	2974488	0,0108163
1KUM-1TUL (108-108)	8 (64)	9677826	0,0351921	9186394	0,0334051
3FIS:A-3FIS:B (98-98)	9 (72)	5474907	0,0199088	6204550	0,0225620
1SCE:A-1PUC (112-105)	12 (96)	-	-	8217428	0,0298816
1CPC:L-1COL:A (172-204)	15 (120)	-	-	109978362	0,3999213
1HBA:A-4HHB:B (141-146)	17 (136)	-	-	73727950	0,2681016
1ATP:E-1KOB:A (350-387)	33 (264)	-	-	262230498	0,9535565

Πίνακας 5.3.2: Κύκλοι προσομοιωμένης εκτέλεσης της υλοποίησης σε FPGA, που έγινε για τα έξι από τα δέκα πειράματα, και εκτιμώμενοι κύκλοι των δέκα πειραμάτων καθώς και οι χρόνοι που τους αντιστοιχούν.

Είσοδοι (Μεγέθη)	AFPs (κατάλοιπα)	Χρόνος εκτέλεσης (sec) FPGA	Εκτιμώμενος χρόνος (sec) FPGA	Bioperf σε Pentium4@2,66GHz (sec)	Μοντελοπίηση σε Pentium4@2,66GHz (sec)
1LYP-1OLG:A (32-42)	3 (24)	0,0005215	0,0004824	0,0113255	0,0070158
1BBT:4-1TMF:4 (85-45)	3 (24)	0,0008341	0,0005071	0,0077170	0,0144325
1TIG-3FIS:A (94-98)	5 (40)	0,0125595	0,0116412	0,0437860	0,2904537
1VIH-1PYT:A (71-94)	7 (56)	0,0114019	0,0108163	0,0320720	0,2704086
1KUM-1TUL (108-108)	8 (64)	0,0351921	0,0334051	0,0736660	0,9729898
3FIS:A-3FIS:B (98-98)	9 (72)	0,0199088	0,0225620	0,0207470	0,5490356
1SCE:A-1PUC (112-105)	12 (96)	-	0,0298816	0,0390880	0,8398902
1CPC:L-1COL:A (172-204)	15 (120)	-	0,3999213	0,5427210	13,2576371
1HBA:A-4HHB:B (141-146)	17 (136)	-	0,2681016	0,1248810	8,3590125
1ATP:E-1KOB:A (350-387)	33 (264)	-	0,9535565	2,0722638	37,3069165

Πίνακας 5.3.3: Σύγκριση χρόνου εκτέλεσης σε FPGA και σε Pentium4@2,66GHz.

Όπως παρατηρείται από την σύγκριση της τρίτης και τέταρτης στήλης ο εκτιμώμενος χρόνος εκτέλεσης του αλγορίθμου σε αναδιατασσόμενη λογική πλησιάζει σε μεγάλο βαθμό τον πραγματικό χρόνο, εκτός από ένα σφάλμα που παρατηρείται στην περίπτωση του δεύτερου πειράματος. Επομένως μπορεί να ληφθεί υπόψη και ο εκτιμώμενος χρόνος στην εκτίμηση της απόδοσης.

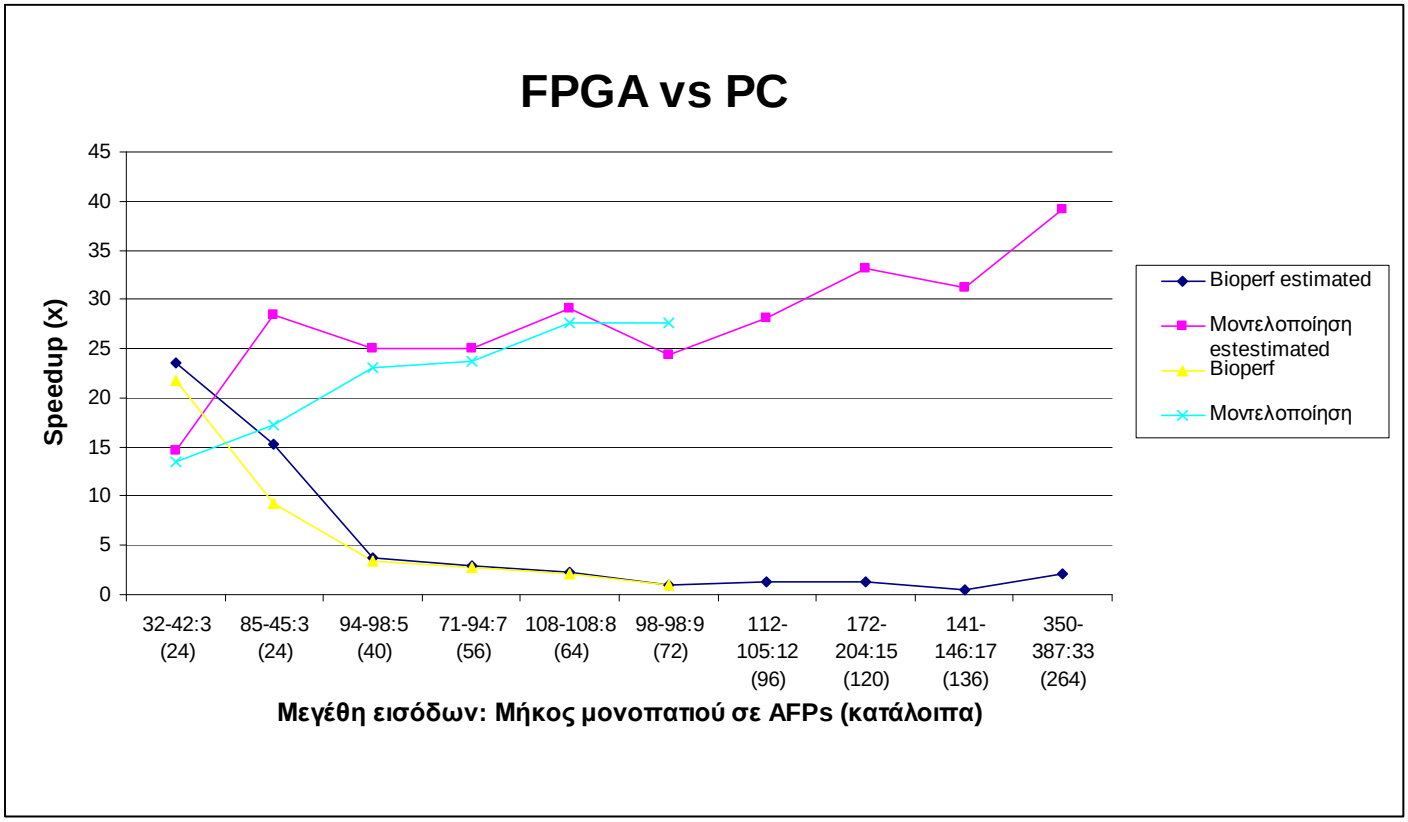
Συγκρίνοντας τον χρόνο εκτέλεσης στην FPGA, λαμβάνοντας υπόψη τόσο τον πραγματικό χρόνο εκτέλεσης όσο και τον εκτιμώμενο, σε σχέση με τον χρόνο εκτέλεσης της εφαρμογής του Bioperf, παρατηρείται ότι όσο αυξάνεται το μέγεθος του βέλτιστου μονοπατιού ομοιότητας μεταξύ των δύο πρωτεϊνικών δομών μειώνεται η απόδοση της FPGA σε σχέση με το PC λόγω του ότι στην εφαρμογή του Bioperf περιλαμβάνεται και ο κώδικας βελτιστοποίησης που σε περιπτώσεις με μεγάλα

μονοπάτια ομοιότητας μειώνει κατά πολύ το εύρος του πεδίου αναζήτησης μειώνοντας τον υπολογιστικό όγκο, αυξάνοντας έτσι την απόδοση του αλγορίθμου. Για τον λόγο αυτό στις περιπτώσεις του έκτου και του ένατου πειράματος όπου συγκρίνονται πρωτεϊνικές δομές με μεγάλη ο χρόνος εκτέλεσης στο PC είναι μικρότερος από τον χρόνο εκτέλεσης σε FPGA. Σε αντίθετη περίπτωση ο χρόνος εκτέλεσης σε FPGA είναι κατά πολύ μικρότερος σε σχέση με το PC.

Κάνοντας την πιο πάνω σύγκριση του χρόνου εκτέλεσης σε FPGA και του χρόνου εκτέλεσης σε PC με ην χρήση της εφαρμογής που δημιουργήθηκε κατά την μοντελοποίηση, στην οποία δεν περιλαμβάνεται η βελτιστοποίηση, παρατηρείται ότι η εκτέλεση σε FPGA υπερτερεί συνεχώς στον χρόνο εκτέλεσης σε σχέση με το PC, έχοντας μία συνεχή βελτίωση όσο αυξάνεται το μέγεθος του βέλτιστου μονοπατιού ομοιότητας.

Είσοδοι (Μεγέθη)	AFPs (κατάλοιπα)	Bioperf Speedup (x) FPGA vs PC	Μοντελοποίηση Speedup (x) FPGA vs PC	Bioperf Εκτιμώμενο Speedup (x) FPGA vs PC	Μοντελοποίηση Εκτιμώμενο Speedup (x) FPGA vs PC
1LYP-1OLG:A (32-42)	3 (24)	21,72	13,45	23,48	14,54
1BBT:4-1TMF:4 (85-45)	3 (24)	9,25	17,30	15,22	28,46
1TIG-3FIS:A (94-98)	5 (40)	3,49	23,13	3,76	24,95
1VIH-1PYT:A (71-94)	7 (56)	2,81	23,71	2,97	25,00
1KUM-1TUL (108-108)	8 (64)	2,09	27,65	2,21	29,13
3FIS:A-3FIS:B (98-98)	9 (72)	1,04	27,58	0,92	24,33
1SCE:A-1PUC (112-105)	12 (96)	-	-	1,31	28,11
1CPC:L-1COL:A (172-204)	15 (120)	-	-	1,36	33,15
1HBA:A-4HHB:B (141-146)	17 (136)	-	-	0,47	31,18
1ATP:E-1KOB:A (350-387)	33 (264)	-	-	2,17	39,12

Πίνακας 5.3.4: Επιταχύνσεις του χρόνου εκτέλεσης των πειραμάτων από την FPGA σε σχέση με το PC όπου εκτελέστηκαν τα πειράματα.



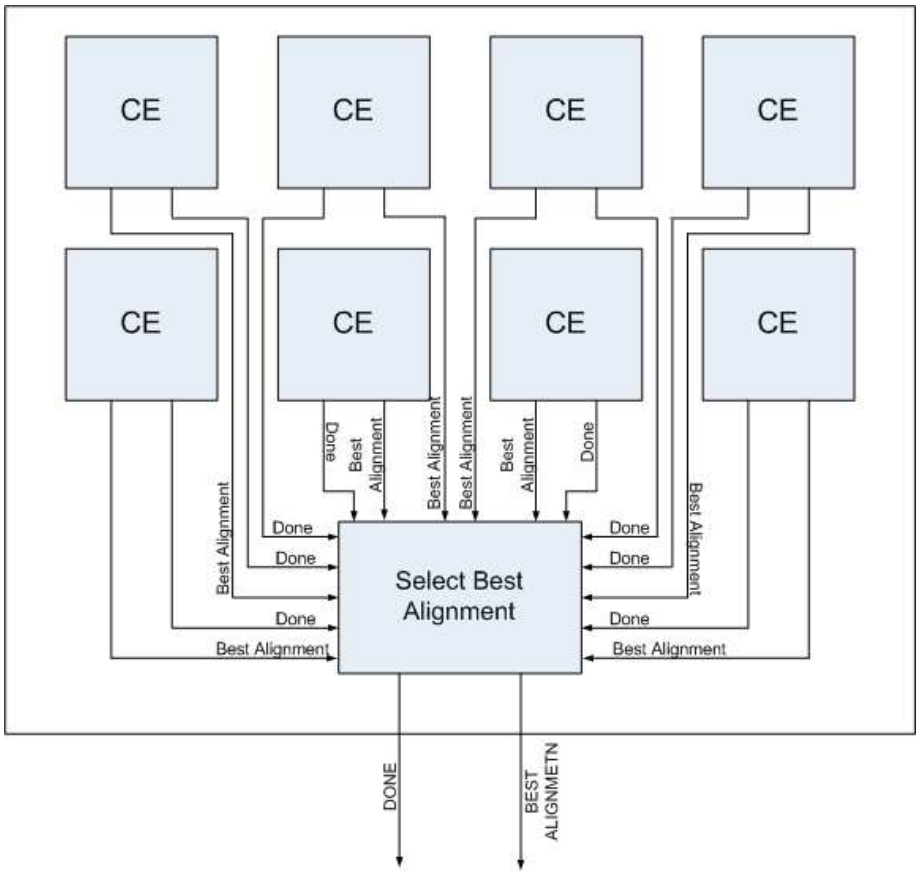
Εικόνα 5.3: Διάγραμμα με τις επιταχύνσεις του χρόνου εκτέλεσης των πειραμάτων από την FPGA σε σχέση με το PC όπου εκτελέστηκαν τα πειράματα.

5.4 Αξιοποίηση των αχρησιμοποίητων πόρων της FPGA με στόχο την βελτίωση της απόδοσης

Ως μια προσπάθειας βελτίωσης της απόδοσης μπορεί να θεωρηθεί η χρησιμοποίηση περισσοτέρων από μία μονάδες του συστήματος που υλοποιήθηκε με στόχο τον ταχύτερο υπολογισμό μίας εξόδου.

Στην FPGA που επιλέχτηκε υπάρχει η δυνατότητα για να χρησιμοποιηθούν παράλληλα οκτώ μονάδες του συστήματος υλοποίησης του αλγορίθμου CE που σχεδιάστηκε και υλοποιήθηκε. Μία σχεδίαση που μπορεί να κάνει χρήση αυτής της δυνατότητας που παρέχεται, είναι πέρα από της οκτώ μονάδες να χρησιμοποιηθεί μία λογική στην οποία όταν το σήμα εξόδου Done και των οκτώ μονάδων είναι 1, δηλαδή όταν ολοκληρωθεί η διαδικασία σε όλες τις μονάδες, να συγκρίνονται οι οκτώ έξοδοι Best_Alignment των μονάδων και να επιλέγεται η βέλτιστη ως έξοδος του συνολικού συστήματος. Επίσης χρειάζεται μία μετατροπή στην κατάσταση Fisrt_AFP της μηχανής πεπερασμένων καταστάσεων της κάθε μονάδας έτσι ώστε να γίνει μία

“δίκαιη” ανάθεση των μονοπατιών που θα εξετάσει η κάθε μονάδα με στόχο το συγχρονισμό του τερματισμού και των οκτώ μονάδων στον βαθμό που αυτό μπορεί να είναι δυνατό. Με τον τρόπο αυτό μπορεί να θεωρηθεί ότι το σύστημα θα εκτελεί την όλη διαδικασία οκτώ φορές γρηγορότερα σε σχέση με την χρησιμοποίηση μόνο μίας μονάδας.



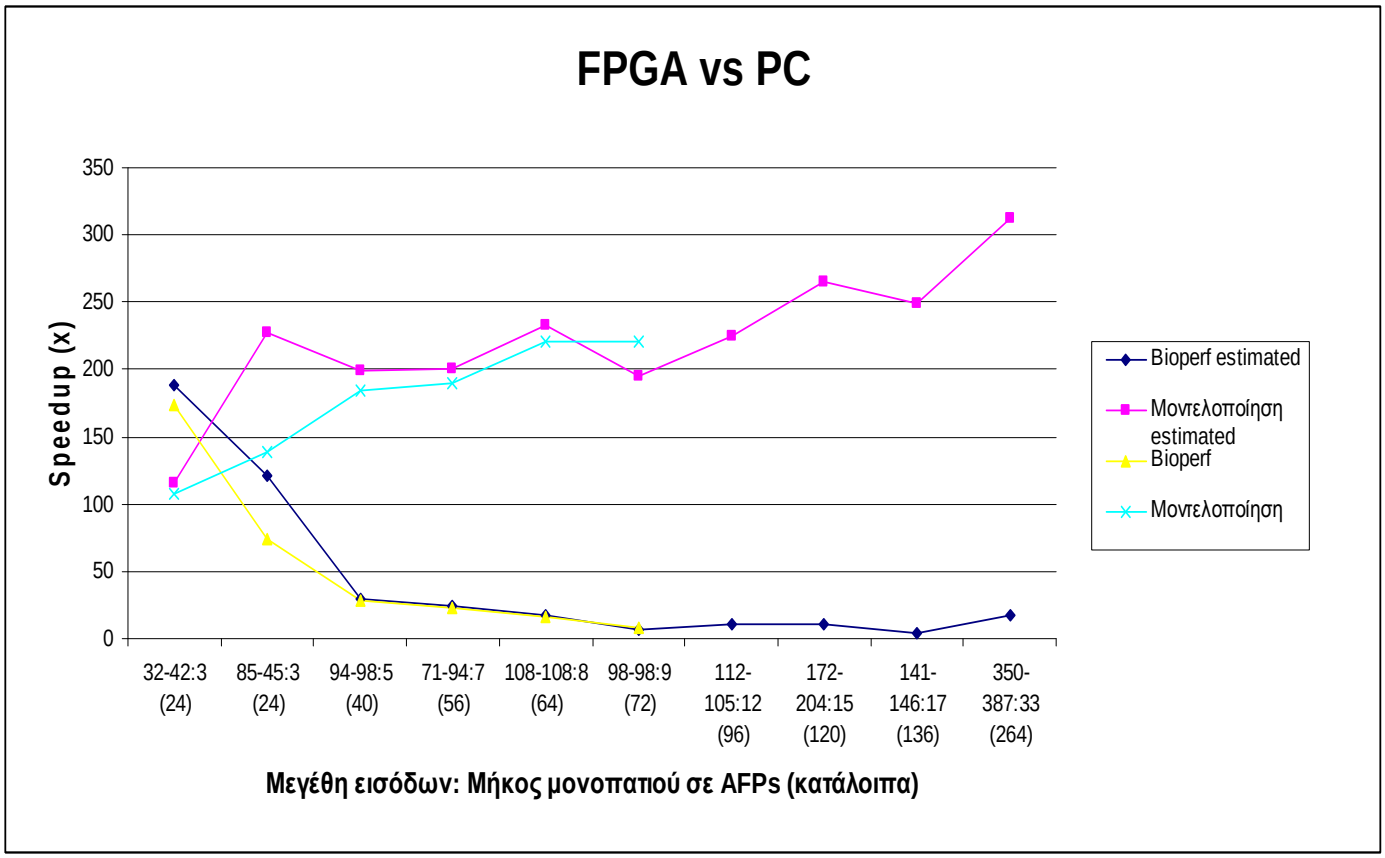
Σχήμα 5.4: Διάταξη των οκτώ μονάδων οι έξοδοι των οποίων λαμβάνονται από την μονάδα Select Best Alignment για να επιλεγεί το βέλτιστο μονοπάτι ομοιότητας.

Device Utilization Summary (xc5vsx240t-2ff1738)			
	Used	Available	Utilization
Number of Slice Registers	136435	149760	91%
Number of Slice LUTs	109304	149760	72%
Number of Slice LUT-FF pairs	78028	167711	46%
Number of bonded IOBs	266	960	27%
Number of Block RAM/FIFO	284	516	55%
Number of BUFG/BUFGCTRLs	16	32	50%
Number of DSP48Es	512	1056	48%

Πίνακας 5.4.1: Χρησιμοποίηση πόρων της FPGA για την υλοποίηση της αρχιτεκτονικής.

Είσοδοι (Μεγέθη)	AFPs (κατάλοιπα)	Bioperf Speedup (x) FPGA vs PC	Μοντελοποίηση Speedup (x) FPGA vs PC	Bioperf Εκτιμώμενο Speedup (x) FPGA vs PC	Μοντελοποίηση Εκτιμώμενο Speedup (x) FPGA vs PC
1LYP-1OLG:A (32-42)	3 (24)	173,76	107,60	187,84	116,32
1BBT:4-1TMF:4 (85-45)	3 (24)	74,00	138,40	121,76	227,68
1TIG-3FIS:A (94-98)	5 (40)	27,92	185,04	30,08	199,60
1VIH-1PYT:A (71-94)	7 (56)	22,48	189,68	23,76	200,00
1KUM-1TUL (108-108)	8 (64)	16,72	221,20	17,68	233,04
3FIS:A-3FIS:B (98-98)	9 (72)	8,32	220,64	7,36	194,64
1SCE:A-1PUC (112-105)	12 (96)	-	-	10,48	224,88
1CPC:L-1COL:A (172-204)	15 (120)	-	-	10,88	265,20
1HBA:A-4HHB:B (141-146)	17 (136)	-	-	3,76	249,44
1ATP:E-1KOB:A (350-387)	33 (264)	-	-	17,36	312,96

Πίνακας 5.3.4: Επιταχύνσεις του χρόνου εκτέλεσης των πειραμάτων με την χρήση οκτώ μονάδων της αρχιτεκτονικής που υλοποιήθηκε.



Εικόνα 5.4: Διάγραμμα με τις επιταχύνσεις του χρόνου εκτέλεσης των πειραμάτων με την χρήση οκτώ μονάδων της αρχιτεκτονικής που υλοποιήθηκε σε σχέση με το PC.

Όπως παρατηρείται στις εικόνες 5.3 και 5.4 δεν υπάρχει μία ξεκάθαρη εικόνα για την επιτάχυνση που επιτυγχάνεται με την υλοποίηση του μέρους του αλγορίθμου CE σε αναδιατασσόμενη λογική σε σχέση με την εφαρμογή του Bioperf. Το γεγονός αυτό οφείλεται στο ότι η εφαρμογή του Bioperf περιλαμβάνει την υλοποίηση της σχέσης, που απορρίπτοντας κάποια μονοπάτια ομοιότητας πριν την ολοκλήρωσή τους, μειώνει τον υπολογιστικό όγκο των πειραμάτων η όποια δεν υλοποιήθηκε σε αναδιατασσόμενη λογική. Αυτό έχει ως αποτέλεσμα σε περιπτώσεις πολύ μικρών μονοπατιών ομοιότητας όπου η βελτιστοποίηση αυτή δεν επιφέρει κάποια σημαντική βελτίωση στον αλγόριθμό να υπάρχει μία σημαντική βελτίωση στον χρόνο εκτέλεσης του αλγορίθμου με την χρήση της υλοποίησης σε αναδιατασσόμενη λογική.

Λαμβάνοντας υπόψη και σταθερή και κατά κάποιο ποσοστό αυξανόμενη, όσο μεγαλώνει το μονοπάτι ομοιότητας, επιτάχυνση της αναδιατασσόμενης λογικής σε σχέση με την εφαρμογή της μοντελοποίησης εκτιμάται ότι η πραγματική επιτάχυνση

σε σχέση με την εφαρμογή του Bioperf κυμαίνεται στα επίπεδα της επιτάχυνσης που παρατηρείται για τα πολύ μικρά μονοπάτια ομοιότητας.

6

Συμπεράσματα και Μελλοντικές Επεκτάσεις και Αλλαγές

Σε αυτό το κεφάλαιο αναφέρονται τα συμπεράσματα και κάποιες ιδέες για αλλαγές ή επεκτάσεις της αρχιτεκτονικής.

6.1 Συμπεράσματα

Η εργασία αυτή ασχολήθηκε με την μελέτη του αλγορίθμου CE και την δημιουργία του κυριότερου μέρους του, που αφορά τον υπολογισμό όλων των μονοπατιών ομοιότητας με στόχο την εύρεση του μεγαλύτερου μονοπατιού ομοιότητας ως προς την τριτοταγή δομή, σε αναδιατασσόμενη λογική. Ο στόχος ήταν το σύστημα αυτό να είναι αποδοτικότερο σε σχέση με ένα PC, στην συγκεκριμένη περίπτωση ένα Pentium4@2,66GHz.

Όπως παρατηρήθηκε από τα αποτελέσματα ο στόχος εκτιμάται ότι ικανοποιείται σε πολύ μεγάλο βαθμό χωρίς αυτό να μπορεί να ειπωθεί με σιγουριά αν δεν προστεθεί στην υλοποίηση σε αναδιατασσόμενη λογική και η βελτιστοποίηση που περιλαμβάνεται στην υλοποίηση του Bioperf

6.2 Μελλοντικές επεκτάσεις και αλλαγές

Η αρχιτεκτονική που σχεδιάστηκε, υλοποιήθηκε και παρουσιάστηκε σε αυτή την εργασία έχει δυνατότητα για να υποστεί αλλαγές που θα βελτιώσουν την απόδοση, αλλά και επεκτάσεις για μία πιο ολοκληρωμένη προσέγγιση του αλγορίθμου CE.

Αλλαγές και επεκτάσεις που μπορούν να πραγματοποιηθούν είναι:

- Αλλαγή της αρχιτεκτονικής έτσι ώστε ένα ο υπολογισμός ενός αθροίσματος να χρειάζεται ένα κύκλο για να γίνει και όχι όσους κύκλους όσες και οι τιμές που προσθέτει το άθροισμα. Με τον τρόπο αυτό αν το συγκεκριμένο άθροισμα αθροίζει 8 τιμές θα επιτύχει μία βελτίωση που θα κάνει το σύστημα 8 φορές γρηγορότερο.
- Επέκταση της αρχιτεκτονικής για πλήρη υλοποίηση του αλγορίθμου και πρωτίστως των συναρτήσεων που μπορούν να θεωρηθούν ως βελτιστοποιήσεις του αλγορίθμου και πρόκειται να βελτιώσουν της απόδοση της υλοποίησης και να δώσουν μία πιο ξεκάθαρη εικόνα για το μέγεθος της βελτιστοποίηση που επιτυγχάνεται με την υλοποίηση του αλγορίθμου σε αναδιατασσόμενη λογική.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Ilya N. Shindyalov and Philip E. Bourne: *"Protein Structure Alignmetn by Incremental Combinatorial Extension (CE) of the Optimal Path"*.
   San Diego Supercomputer Center, PO Box 85608, San Diego, CA 92186, USA.
   Department of Pharmacology, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA.
   The Burnham Institute, 10901 North Torrey Pines Road, La Jolla, CE 92036, USA.
   Wednesday, February 25, 1998 – final.
   (http://cl.sdsc.edu/ce/ce.pdf).

2. The Contents of a PDB File
   (http://spdbv.vital-it.ch/TheMolecularLevel/0Help/PDBContent.html)

3. Philip E. Bourne: *"Reductionism and Classification Require Detailed Comparison Consider 3D Comparison"*.
   Pharm 201/Bioinformatics I, Department of Pharmacology, UCSD.
   (www.sdsc.edu/pb/edu/pharm201/11/11.ppt)

4. Protein Structure
   (bioinfo.tau.ac.il/~intro_bioinfo/Exercises/**6_proteinStructure.ppt**)

5. ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ: Υπολογιστική Βιολογία
   (www.cs.ucy.ac.cy/courses/EPL450/lectures/**lecture1new**.ppt)

6. The Importance of Protein Structure
   (www.elmhurst205.org/staff/kvanhoeck/genetics/**prostruct**.ppt)

7. Γ. ΤΡΑΝΟΣ, ΒΙΟΛΟΓΙΑ ΓΕΝΙΚΗΣ ΠΑΙΔΕΙΑΣ,
   3^{ο} Ε.Λ. ΑΡΓΥΡΟΥΠΟΛΗΣ

8. Η. ΗΛΙΟΠΟΥΛΟΣ, ΜΟΡΙΑΚΗ ΑΝΑΓΝΩΡΙΣΗ, ΓΠΑ, 2001

9. Carl Branden & John Tooze, Εισαγωγή στη Δομή των Πρωτεϊνών (Δεύτερη Έκδοση), Ακαδημαϊκές Εκδόσεις Μπάσδρα, 2006

10. Χημεία του Κυττάρου Βιολογικά Μακρομόρια (www.materials.uoc.gr/courses/ETY492/notes/**macromol.pdf**)

11. Liisa Holm and Chris Sander: *"Protein Structure Comparison by Alignment of Distance Matrices"*.
Protein Design Group, European Molecular Biology Laboratory, D-69012 Heidelberg, Federal Republic of Germany.
30 April 1993.
(http://www.gersteinlab.org/courses/452/09-spring/pdf/Holm.pdf)

12. Tolga Can and Yuan-Fang Wang: *"CTSS: A Robust and Efficient Method for Protein Structure Alignment"*.
Department of Computer Science, University of California at Santa Barbara, USA.
(http://conferences.computer.org/bioinformatics/csb2003/PDF/reg_pprs/015-can_CSB03.pdf)

13. Dali: an algorithm for optimal structure alignment using distance matrices (http://cgi.di.uoa.gr/~piv07004/Dali.pdf)

14. SAN DIEGO SUPERCOMPUTER CENTER: Combinatorial Extension Addresses Challenges of Structural Genomics (http://www.sdsc.edu/pub/envision/v16.3/ce.html)

15. PADS: Protein structure Alignment using Directional shape Signatures (http://repositories.cdlib.org/postprints/1467/)

16. Invariant structural core found by TOPOFIT protein structure alignment (http://mozart.bio.neu.edu/topofit/index.php)

17. FAST: A novel protein structure alignment algorithm (http://www.ncbi.nlm.nih.gov/pubmed/15609341)